



TESIS - SM 142501

**APLIKASI METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK  
KLASIFIKASI TANAMAN BERDAUN MENJARI DAN GULMA  
BERDASARKAN FITUR BENTUK DAN TEKSTUR DAUN**

ETRIANA MEIRISTA  
NRP 1213 201 044

DOSEN PEMBIMBING  
Dr. Imam Mukhlash, S.Si., M.T.  
Dr. Budi Setiyono, S.Si., M.T.

PROGRAM MAGISTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2015



THESIS - SM 142501

**THE APPLICATION OF SUPPORT VECTOR MACHINE (SVM)  
METHOD FOR CLASSIFY LEAVES FINGER PLANTS AND WEEDS  
BASED ON LEAVES SHAPE AND TEXTURE FEATURES**

ETRIANA MEIRISTA  
NRP 1213 201 044

SUPERVISOR  
Dr. Imam Mukhlash, S.Si., M.T.  
Dr. Budi Setiyono, S.Si., M.T.

MASTER'S DEGREE  
MATHEMATICS DEPARTMENT  
FACULTY OF MATHEMATICS AND NATURAL SCIENCES  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2015



**APLIKASI METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK  
KLASIFIKASI TANAMAN BERDAUN MENJARI DAN GULMA  
BERDASARKAN FITUR BENTUK DAN TEKSTUR DAUN**

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar Magister  
Sains (M.Si.)

di

Institut Teknologi Sepuluh Nopember Surabaya

Oleh :

ETRIANA MEIRISTA

NRP. 1213 201 044


Tanggal Ujian : 03 Juli 2015

Periode Wisuda : September 2015


Disetujui Oleh :

  
Dr. Imam Mukhlash, S.Si, M.T.  
NIP. 19700831 199403 1 003

(Pembimbing I)

  
Dr. Budi Setiyono, S.Si, M.T.  
NIP. 19720207 199702 1 001

(Pembimbing II)


  
Prof. Dr. Mohammad Isa Irawan, M.T.  
NIP. 19631225 198903 1 001

(Penguji)

  
Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T.  
NIP. 19690405 199403 2 003

(Penguji)

Direktur Program Pascasarjana,

  
Prof. Dr. Ir. Adi Soeprijanto, M.T.  
NIP. 19640405 199002 1 001

PASCASARJANA



# **APLIKASI METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK KLASIFIKASI ANTARA TANAMAN BERDAUN MENJARI DAN GULMA BERDASARKAN FITUR BENTUK DAN TEKSTUR DAUN**

Nama Mahasiswa : Etriana Meirista  
NRP : 1213 201 044  
Dosen Pembimbing : 1. Dr. Imam Mukhlash, S.Si., M.T.  
2. Dr. Budi Setiyono, S.Si., M.T.

## **ABSTRAK**

Dewasa ini, berbagai cara dilakukan untuk meningkatkan hasil produksi pertanian. Salah satunya penggunaan herbisida untuk membasmi gulma. Namun, ada beberapa gulma yang memiliki kemiripan dengan tanaman, sehingga diperlukan klasifikasi antara tanaman dan gulma sebelum melakukan pembasmian menggunakan herbisida. Tanaman semangka merupakan objek pada penelitian ini. Klasifikasi tanaman berdasarkan kemiripan daun yang dimiliki tanaman tersebut menggunakan citra digital yang dibagi menjadi tiga tahap. Pada tahap pertama, *preprocessing* dengan melakukan proses *cropping* citra, *resize* citra memisahkan *background* dan *foreground*, selanjutnya dilakukan segmentasi deteksi tepi dengan menggunakan operator Canny. Tahap kedua dilakukan ekstraksi fitur untuk mengambil informasi-informasi penting untuk pengenalan fitur daun. Fitur yang digunakan adalah fitur bentuk dan tekstur. Selanjutnya dilakukan pengklasifikasian daun tersebut sebagai tanaman berdaun menjari dan gulma dengan menggunakan metode *Support Vector Machine* (SVM). Metode SVM terbukti memiliki akurasi yang baik untuk mengklasifikasikan tanaman berdaun menjari dan gulma berdasarkan fitur bentuk dan tekstur pada citra multi daun dengan menggunakan kernel *quadratic*. Rata-rata akurasi yang dimiliki yaitu sebesar 74,54%.

**Kata Kunci:** Deteksi Tepi, Ekstraksi Fitur, Klasifikasi Tanaman, Multi Daun dan SVM.

# **APLIKASI METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK KLASIFIKASI ANTARA TANAMAN BERDAUN MENJARI DAN GULMA BERDASARKAN FITUR BENTUK DAN TEKSTUR DAUN**

Nama Mahasiswa : Etriana Meirista  
NRP : 1213 201 044  
Dosen Pembimbing : 1. Dr. Imam Mukhlash, S.Si., M.T.  
2. Dr. Budi Setiyono, S.Si., M.T.

## **ABSTRAK**

Dewasa ini, berbagai cara dilakukan untuk meningkatkan hasil produksi pertanian. Salah satunya penggunaan herbisida untuk membasmi gulma. Namun, ada beberapa gulma yang memiliki kemiripan dengan tanaman, sehingga diperlukan klasifikasi antara tanaman dan gulma sebelum melakukan pembasmian menggunakan herbisida. Tanaman semangka merupakan objek pada penelitian ini. Klasifikasi tanaman berdasarkan kemiripan daun yang dimiliki tanaman tersebut menggunakan citra digital yang dibagi menjadi tiga tahap. Pada tahap pertama, *preprocessing* dengan melakukan proses *cropping* citra, *resize* citra memisahkan *background* dan *foreground*, selanjutnya dilakukan segmentasi deteksi tepi dengan menggunakan operator Canny. Tahap kedua dilakukan ekstraksi fitur untuk mengambil informasi-informasi penting untuk pengenalan fitur daun. Fitur yang digunakan adalah fitur bentuk dan tekstur. Selanjutnya dilakukan pengklasifikasian daun tersebut sebagai tanaman berdaun menjari dan gulma dengan menggunakan metode *Support Vector Machine* (SVM). Metode SVM terbukti memiliki akurasi yang baik untuk mengklasifikasikan tanaman berdaun menjari dan gulma berdasarkan fitur bentuk dan tekstur pada citra multi daun dengan menggunakan kernel *quadratic*. Rata-rata akurasi yang dimiliki yaitu sebesar 74,54%.

**Kata Kunci:** Deteksi Tepi, Ekstraksi Fitur, Klasifikasi Tanaman, Multi Daun dan SVM.

## KATA PENGANTAR



Assalamualaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirobbil'alamin. Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya. Shalawat dan salam kepada Rasulullah Nabi Muhammad SAW sebagai suri tauladan sehingga penulis dapat menyelesaikan tesis yang berjudul “**APLIKASI METODE *SUPPORT VECTOR MACHINE* (SVM) UNTUK KLASIFIKASI TANAMAN BERDAUN MENJARI DAN GULMA BERDASARKAN FITUR BENTUK DAN TEKSTUR DAUN**” sebagai salah satu persyaratan kelulusan dalam memperoleh gelar Master pada Program Studi PascaSarjana Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tersusunnya Tesis ini tidak terlepas dari bantuan berbagai pihak. Untuk itu penulis ingin menyampaikan terimakasih kepada:

1. Ayahanda H. Nasir, (Almh) Ibunda tercinta Hj. Setyaningsih, Kakanda Edwin dan Erlin yang senantiasa mendoakan, memberikan cinta dan dukungan baik moril maupun material, serta selalu mengingatkan penulis agar selalu berikhtiar, bersyukur, bersabar, dan bertawakkal.
2. Bapak Dr. Imam Mukhlash, S.Si., M.T. dan Bapak Dr. Budi Setiyono, S.Si., M.T. selaku dosen pembimbing yang memberikan bimbingan, arahan, motivasi dan waktu yang telah diluangkan kepada penulis untuk berdiskusi dalam menyelesaikan tesis ini.
3. Bapak Prof. Dr. Mohammad Isa Irawan, M.T. dan Ibu Dr. Dwi Ratna S.Si., M.T. selaku dosen penguji pada seminar hasil dan sidang tesis yang telah memberikan masukan dan saran selama penulis menjalani perkuliahan dan untuk menyempurnakan tesis ini.
4. Bapak DR. Mahmud Yunus, M.Si. selaku dosen wali yang telah memberikan masukan dan saran selama penulis menjalani perkuliahan.
5. Bapak Dr. Subiono M.S. selaku Ketua Program Studi Pascasarjana Matematika ITS dan Ibu Prof. Dr. Erna Apriliani, M.Si. selaku Ketua Jurusan Matematika ITS yang telah memberikan dukungan dan kemudahan dalam penyelesaian tesis.

6. Bapak Prof. Ir. Joni Hermana, M.Sc.ES., Ph.D. selaku Rektor Institut Teknologi Sepuluh Nopember (ITS) Surabaya.
7. Bapak Prof. Dr. Ir. Adi Soeprijanto, M.T. selaku Direktur Program Pascasarjana ITS.
8. Direktorat Jendral Perguruan Tinggi (DIKTI) selaku pemberi dana pendidikan melalui program beasiswa Pra S2-S2 SAINTEK.
9. Bapak/Ibu dosen yang telah memberikan bekal dan ilmu pengetahuan, selama berada di matematika ITS, serta staf administrasi Program Studi Magister Matematika atas segala bantuannya.
10. Teman-teman Pascasarjana Matematika Angkatan 2013: Winda, Rahma, Asra, Amel, Zunif, Reta, Dika, Ato, Wawan, Restu, Devi, Umi, Nisa M, Pohet, Hajar, Nia, Novian, Akbar, Sarif, Andri, Galuh, Gani, Alfian, Petrus, Bima, Ninis, Olif dan semua yang tidak dapat disebutkan satu persatu, terima kasih sudah menorehkan banyak kenangan dan berbagi ilmu selama masa perkuliahan serta telah banyak membantu sehingga tesis ini dapat terselesaikan.
11. Tante Yuni, Om Yoyok, Om Heri, Tante Har, Eko P, Yanuar dan semua pihak yang tidak mungkin penulis sebutkan satu persatu yang telah banyak membantu sehingga tesis ini dapat terselesaikan.

Penulis berharap laporan Tesis ini dapat memberikan manfaat bagi masyarakat dan ilmu pengetahuan. Dalam penyusunannya penulis menyadari bahwa Tesis ini masih jauh dari sempurna, oleh karena itu penulis mengharapkan saran dan kritik yang membangun untuk perbaikan di masa yang akan datang.

Wassalamualaikum Warahmatullahi Wabarakatuh

Surabaya, Juli 2014

**Penulis**

## DAFTAR ISI

<b>HALAMAN JUDUL</b>	
<b>LEMBAR PENGESAHAN</b> .....	i
<b>ABSTRAK</b> .....	iii
<b>ABSTRACT</b> .....	v
<b>KATA PENGANTAR</b> .....	vii
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	xiii
<b>DAFTAR TABEL</b> .....	xv
<b>BAB 1 PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	5
<b>BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI</b> .....	7
2.1 Daun .....	7
2.2 Gulma .....	7
2.3 Citra Warna .....	9
2.4 Segmentasi .....	10
2.3.1 Deteksi Tepi Canny .....	10
2.5 <i>Ekstraksi Fitur</i> .....	12
2.5.1 <i>Fitur Bentuk</i> .....	12
a. <i>Fitur Dasar Geometri</i> .....	12
b. <i>Fitur Morfologi Digital</i> .....	13
2.5.2 <i>Fitur Tekstur</i> .....	14
2.6 Klasifikasi dengan <i>Support Vector Machine</i> .....	16
2.7 Penelitian-penelitian yang relevan .....	24
<b>BAB 3 METODOLOGI PENELITIAN</b> .....	27



3.1 Tahapan Penelitian .....	27
3.1.1 Akuisisi .....	27
3.1.2 <i>Preprocessing</i> .....	27
a. <i>Cropping</i> Citra .....	27
b. <i>Resize</i> Citra .....	27
c. Pemisahan <i>Background</i> dan <i>Foreground</i> Citra serta Menghilangkan Objek yang Tidak Diinginkan .....	27
d. Segmentasi Citra .....	28
3.1.3 Ekstraksi Fitur .....	28
3.1.4 Klasifikasi dengan metode SVM .....	28
3.1.5 Uji Coba Hasil Akurasi Klasifikasi .....	28
3.1.6 Penyusunan Tesis .....	29
3.2 Diagram Proses Penelitian .....	29
<b>BAB 4 ANALISIS DAN PEMBAHASAN</b> .....	33
4.1 <i>Preprocessing</i> .....	33
4.1.1 <i>Cropping</i> Citra .....	34
4.1.2 <i>Resize</i> Citra .....	36
4.1.3 Pemisahan <i>Background</i> dan <i>Foreground</i> Citra serta Menghilangkan Objek yang Tidak Diinginkan .....	40
4.1.4 Segmentasi Citra dengan Menggunakan Deteksi Tepi .....	47
4.2 Ekstraksi Fitur .....	50
4.2.1 Fitur Bentuk .....	50
4.2.2 Fitur Tekstur .....	52
4.3 Klasifikasi dengan Menggunakan Metode SVM .....	63
4.4 Implementasi Proses Klasifikasi dengan Matlab .....	72
4.4.1 Implementasi <i>Preprocessing</i> .....	72
a. Pemisahan <i>Background</i> dan <i>Foreground</i> serta Menghilangkan Objek yang Tidak Diinginkan .....	73
b. Segmentasi Deteksi Tepi dengan Operator <i>Canny</i> .....	74
4.4.2 Implementasi Ekstraksi Fitur .....	74
4.4.3 Implementasi Metode <i>Support Vector Machine</i> (SVM) .....	76



a. Training .....	76
b. Pengambilan Objek .....	77
c. Uji ( <i>Testing</i> ) .....	78
4.5 Uji Coba Program .....	78
4.6 Analisis Hasil .....	81
<b>BAB 5 KESIMPULAN DAN SARAN .....</b>	<b>85</b>
5.1 <i>Kesimpulan</i> .....	85
5.2 Saran .....	85
<b>DAFTAR PUSTAKA .....</b>	<b>87</b>
<b>LAMPIRAN 1 EKSTRAKSI FITUR BENTUK .....</b>	<b>89</b>
LAMPIRAN 1.1 EKSTRAKSI FITUR BENTUK SEMANGKA.....	91
LAMPIRAN 1.2 EKSTRAKSI FITUR BENTUK GULMA.....	93
<b>LAMPIRAN 2 EKSTRAKSI FITUR TEKSTUR.....</b>	<b>95</b>
LAMPIRAN 2.1 EKSTRAKSI FITUR TEKSTUR SEMANGKA.....	100
LAMPIRAN 2.2 EKSTRAKSI FITUR TEKSTUR GULMA.....	104
<b>LAMPIRAN 3 DATA CITRA DAUN SEMANGKA .....</b>	<b>107</b>
<b>LAMPIRAN 4 DATA CITRA DAUN GULMA.....</b>	<b>115</b>
<b>LAMPIRAN 5 CITRA MULTI DAUN.....</b>	<b>117</b>
<b>LAMPIRAN 6 KLASIFIKASI MENGGUNAKAN METODE     SUPPORT VEKTOR MACHINE .....</b>	<b>119</b>



## DAFTAR TABEL

Tabel 2.1 <i>Cooccurrence Features</i> .....	15
Tabel 4.1 Contoh Data yang Menggunakan Metode <i>Support Vector Machine</i> .....	64
Tabel 4.2 Hasil Klasifikasi dengan Metode SVM da Nilai Akurasi .....	82



## DAFTAR GAMBAR

Gambar 2.1 Citra Kemiripan Daun Tanaman Semangka dan Gulma.....	8
Gambar 2.2 Diagram Arah Gradien dalam Operator Canny.....	11
Gambar 2.3 Hubungan Panjang Fisiologis dan Lebar Fisiologis.....	13
Gambar 2.4 Sudut dalam GLCM .....	15
Gambar 2.5 Margin Kecil dan Margin Besar.....	17
Gambar 2.6 Sebaran Data yang Terpisah Secara Nonlinier.....	21
Gambar 3.1 Rancangan klasifikasi daun.....	29
Gambar 3.2 Diagram Alir Proses <i>Training</i> SVM .....	30
Gambar 3.3 Diagram Alir Proses <i>Testing</i> SVM .....	31
Gambar 3.4 Blok Diagram Desain Sistem .....	32
Gambar 4.1 Citra Asli .....	33
Gambar 4.2 Proses <i>Cropping</i> Citra Berdasarkan Titik Koordinat .....	34
Gambar 4.3 <i>Cropping</i> Citra Berdasarkan Jumlah Piksel .....	35
Gambar 4.4 Proses <i>Cropping</i> Citra Berdasarkan Jumlah Piksel.....	36
Gambar 4.5 Hasil <i>Cropping</i> Citra .....	36
Gambar 4.6 <i>Resize</i> Citra.....	39
Gambar 4.7 (a) Matriks Hitam Putih Komponen <i>Hue</i> , (b) Citra Hitam Putih Komponen <i>Hue</i> .....	43
Gambar 4.8 (a) Matriks Hitam Putih Komponen <i>Saturation</i> , (b) Citra Hitam Putih Komponen <i>Hue</i> .....	44
Gambar 4.9 Hasil Pemisahan <i>Background</i> dan <i>Foreground</i> untuk Citra Berukuran 10 x 10.....	45
Gambar 4.10 Proses Pemisahan <i>Background</i> dan <i>Foreground</i> .....	46



Gambar 4.11 Penggabungan Citra Hitam Putih dari komponen Warna <i>Hue</i> dan <i>Saturation</i> .....	47
Gambar 4.12 Segmentasi dengan Metode Deteksi Tepi Operator <i>Canny</i> ..	47
Gambar 4.13 Deteksi Tepi Berukuran 10 x 10 .....	50
Gambar 4.14 Proses Ekstraksi Bentuk .....	51
Gambar 4.15 Matriks Citra Awal untuk Perhitungan GLCM.....	52
Gambar 4.16 Tampilan Awal GUI.....	79
Gambar 4.17 Proses <i>Training</i> .....	79
Gambar 4.18 Proses Pemilihan Data Uji.....	80
Gambar 4.19 Proses Klasifikasi Citra Multi Daun .....	80



# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Peningkatan hasil pertanian dan pengoptimalan modal awal banyak dilakukan oleh petani. Sebagian besar petani hanya memperkirakan berapa jumlah herbisida yang akan digunakan untuk membasmi gulma dalam rangka meningkatkan kualitas tanaman yang dimiliki. Gulma adalah tumbuhan yang tidak dikehendaki dalam budidaya yang dilakukan. Objek penelitian ini adalah tanaman semangka, dengan demikian yang dianggap gulma adalah semua tumbuhan yang tumbuh selain tanaman semangka. Misalkan melon yang tumbuh di sela-sela tanaman monokultur semangka dapat dianggap sebagai gulma, namun pada sistem tumpang sari keduanya merupakan tanaman utama. Meskipun demikian, beberapa jenis tumbuhan dikenal sebagai gulma utama, seperti rumput, teki gulma berdaun lebar, dan alang-alang. Daun dari varietas melon ada yang memiliki kemiripan bentuk dengan daun varietas tanaman semangka. Terdapat pula gulma yang memiliki bentuk yang sama seperti daun tanaman semangka yang tidak dapat dibedakan dengan kasat mata.

Herbisida adalah senyawa yang disebarkan pada lahan pertanian untuk menekan atau memberantas gulma yang menyebabkan penurunan hasil pertanian. Herbisida selain memiliki keuntungan juga memiliki kerugian. Kerugian yang disebabkan oleh herbisida ini dapat berdampak pada manusia, tanaman maupun lingkungan berupa *residual effect* dan pencemaran lingkungan, penumpukan biologis, dan tanaman yang bukan merupakan sasaran ikut terbunuh. Oleh karena itu, mesin digital hadir untuk mempermudah mengklasifikasikan antara tanaman pertanian dan gulma sehingga dapat diaplikasikan untuk pengoptimalan penggunaan herbisida pada lahan yang dimiliki petani.

Tanaman pada dasarnya diklasifikasikan berdasarkan bentuk daun dan bunga. Daun memiliki bentuk dua dimensi dan bunga berbentuk tiga dimensi yang memiliki struktur yang kompleks sehingga sulit untuk menganalisis bentuk dan struktur bunga (Wang dkk, 2008). Pengklasifikasian tanaman berdasarkan daun



dengan cara konvensional membutuhkan waktu yang cukup lama. Oleh karena itu, sangat penting mengembangkan sistem yang akurat, cepat, dan efisien dalam mengidentifikasi berbagai spesies tanaman dan mengklasifikasikannya.

Penelitian tentang identifikasi tanaman berdasarkan daun telah banyak dilakukan. Perancangan sistem pengenalan tanaman yang otomatis diperlukan dan berguna karena dapat memudahkan klasifikasi, pemahaman, dan mengelola tanaman secara cepat (Du, dkk., 2007). Arribas, dkk. (2011) mengklasifikasikan daun dari tanaman bunga matahari dengan menggunakan visi komputer dan *neural network*. Tujuan utama dari penelitian ini untuk membedakan antara bunga matahari dan gulma berdasarkan ruang warna menggunakan *Generalized Softmax Perceptron* (GSP) arsitektur jaringan saraf yang *di-training* dengan bantuan algoritma *Posterior Probability Model Selection* (PPMS).

Fokus penelitian mengenai metode klasifikasi tanaman berdasarkan bentuk daun adalah tentang ekstraksi fitur daun dan desain metode klasifikasi yang digunakan. Data daun untuk *testing* seringkali menggunakan Gambar *outdoor* atau *indoor* yang memiliki latar belakang rumit, sederhana atau tanpa latar belakang. Wang, dkk. (2008) mengajukan *framework* klasifikasi yang efektif pada citra daun yang memiliki latar belakang rumit dimana beberapa gangguan dan fenomena *overlapping* mungkin terjadi. Segmentasi citra tanaman yang diambil pada daerah *outdoor* masih merupakan masalah bagi penelitian pertanian. Kondisi pencahayaan yang bervariasi, bayangan tanaman dan sudut matahari dapat berdampak langsung pada citra yang diambil. Oleh karena itu, dibutuhkan metode yang dapat membedakan antara kelas piksel tanaman dan yang bukan kelas piksel tanaman untuk mengurangi dampak dari *noise* dan menjaga detail obyek lokal dari citra. Bai, dkk. (2014) mengajukan metode segmentasi vegetasi baru berdasarkan *Particle Swarm Optimization* (PSO) *Clustering* dan model morfologi pada ruang warna yang dapat menangani permasalahan tersebut. Langkah awal dari metode tersebut yaitu menentukan jumlah *cluster*, kemudian *tool* dari dilasi dan erosi morfologi bekerja untuk membentuk model vegetasi warna.

Metode *Support Vector Machine* (SVM) banyak digunakan peneliti untuk identifikasi dan klasifikasi karena memiliki tingkat akurasi yang tinggi.



*Training SVM* dengan dataset yang besar sangat kompleks. *Decision tree* digunakan untuk mendeteksi entropi *region* yang rendah dalam ruang *input* dan diskriminan linier *Fisher* diaplikasikan untuk mendeteksi data yang dekat dengan *support vector* (Chau, dkk., 2014). *SVM multi-class* dengan arsitektur pohon biner dapat diaplikasikan untuk mengidentifikasi aktifitas manusia (Qian, dkk., 2010). Metode SVM juga digunakan oleh Narayan dan Subbrayan (2014) untuk mengklasifikasikan tanaman dan pohon berdasarkan citra daun tunggal dengan mengoptimalkan seleksi subset fitur. Konsep SVM secara sederhana dinyatakan sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah class pada *input space*. Masalah klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok.

Beberapa penelitian tentang fitur tekstur, fitur bentuk, fitur warna, fitur vena dan tepi daun telah dilakukan (Gwo, dkk., 2013; Devi dan Pallavi, 2014). Gwo, dkk. (2013) menggunakan teorema *Bayes* untuk pencocokan *rotary* dalam mengidentifikasi tepi daun tanaman. Sedangkan Devi dan Pallavi (2014) memberikan kerangka baru untuk mengenali dan mengidentifikasi tanaman berdasarkan fitur bentuk, vena, warna dan tekstur.

Pada penelitian ini metode SVM digunakan untuk mengklasifikasikan tanaman berdaun menjari dan gulma berdasarkan fitur bentuk dan tekstur daun pada sebuah citra yang di dalamnya terdapat lebih dari satu daun. Bagian dari tanaman yang diamati adalah daun karena memiliki struktur yang lebih sederhana. Daun juga mudah ditemukan dan dikumpulkan pada setiap musim. Tahapan yang dilakukan pada penelitian ini adalah *cropping* dan *resize* citra, selanjutnya pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan. Proses pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan dilakukan dengan mengubah citra *Red Green Blue* (RGB) menjadi *Hue Saturation Value* (HSV). Komponen intensitas yang digunakan hanya *Hue* dan *Saturation*. Setelah itu dilakukan proses segmentasi, ekstraksi fitur bentuk dan tekstur. Fitur bentuk yang diekstraksi merupakan fitur morfologi digital yang menggunakan fitur dasar geometri dan yang meliputi diameter, panjang fisiologis, lebar fisiologis, luas daun, perimeter, aspek rasio, faktor *form*, *rectangularity*, faktor *narrow*, rasio perimeter dari daun, rasio

perimeter dari panjang fisiologis dan lebar fisiologis. Sedangkan fitur tekstur akan diekstraksi dengan menggunakan *Gray Level Co-occurrence Matrices* (GLCM).

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang yang ada, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana penerapan metode *Support Vector Machine* (SVM) dalam mengklasifikasikan tanaman berdaun menjari dan gulma berdasarkan daun?
- b. Bagaimana menganalisis hasil uji coba terkait dengan akurasi dari aplikasi yang telah dibuat?

## **1.3 Batasan Masalah**

Untuk membatasi kajian agar lebih spesifik dan terarah, maka penelitian ini dibatasi pada pembahasan sebagai berikut:

- a. Objek yang digunakan adalah daun dari tanaman semangka yang berasal dari pertanian di kabupaten Nganjuk Provinsi Jawa Timur.
- b. Diasumsikan citra yang digunakan tidak memiliki *oklusi* dan tumpang tindih.
- c. Data set yang digunakan adalah citra daun semangka berumur 15 hari dan 40 hari.
- d. Objek pada foreground selain daun dihilangkan.
- e. Implementasi program menggunakan program MATLAB.

## **1.4 Tujuan Penelitian**

Dari perumusan masalah maka dapat dirumuskan tujuan penelitian ini sebagai berikut:

- a. Mendapatkan hasil klasifikasi tanaman berdaun menjari dan gulma berdasarkan daun dengan menggunakan metode *Support Vector Machine* (SVM).
- b. Menganalisis hasil dengan melakukan uji coba terhadap perangkat lunak untuk mengetahui akurasi.



### **1.5 Manfaat Penelitian**

Bagi masyarakat khususnya petani dan ahli botani, penelitian ini dapat menambah pengetahuan mengenai klasifikasi tanaman berdaun menjari dan gulma dengan menggunakan citra multi daun digital yang diterapkan ke metode *Support Vector Machine* (SVM) sehingga dapat diaplikasikan dalam pengoptimalan penggunaan herbisida.

## **BAB 2**

### **KAJIAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Daun**

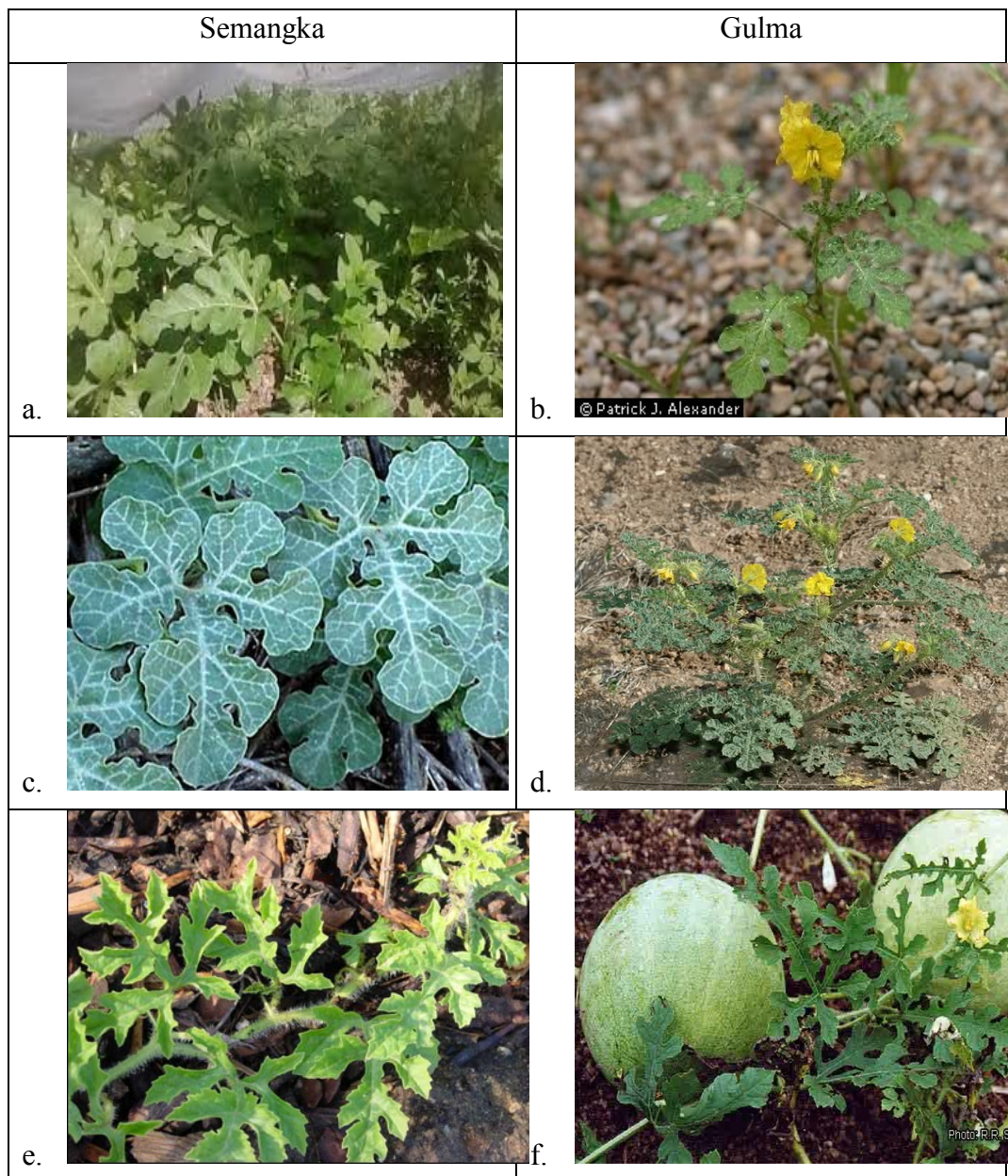
Daun merupakan salah satu bagian penting dari tanaman karena memiliki klorofil, yaitu tempat terjadinya proses pembentukan makanan (karbohidrat/gula) dari CO<sub>2</sub> dengan energi matahari atau lebih dikenal dengan fotosintesis. Pada umumnya daun berbentuk pipih bilateral, berwarna hijau, dan merupakan tempat utama terjadinya proses fotosintesis (Ratnasari, 2008).

#### **2.2 Gulma**

Gulma didefinisikan sebagai tumbuhan yang tumbuh di tempat yang tidak dikehendaki. Menurut definisi ini, tidak ada batasan yang jelas antara gulma dan tanaman budidaya, dalam arti setiap tanaman yang tumbuh di tempat yang tidak dikehendaki dikategorikan sebagai gulma (Wahyudi dkk, 2008). Pada tingkat tertentu, tanaman berguna dapat menjadi gulma. Sebaliknya, tumbuhan yang biasanya dianggap gulma dapat pula dianggap tidak mengganggu. Sebagai contoh, melon yang tumbuh di sela-sela pertanaman monokultur semangka dapat dianggap sebagai gulma, namun pada sistem tumpang sari keduanya merupakan tanaman utama. Meskipun demikian, beberapa jenis tumbuhan dikenal sebagai gulma utama, seperti rumput, teki gulma berdaun lebar dan alang-alang

Daun dari varietas melon ada yang memiliki kemiripan bentuk dengan daun varietas tanaman semangka (Gambar 1. (e) dan (f)). Terdapat pula gulma yang memiliki bentuk yang sama seperti daun tanaman semangka yang tidak dapat dibedakan dengan kasat mata. Gulma ini bernama *buffalobur nightshade*. Hal ini akan lebih sulit dibedakan apabila gulma belum mengeluarkan duri. Pada gambar 2.1, diberikan contoh citra gulma yang memiliki kemiripan bentuk dengan tanaman semangka.





Gambar 2.1 Citra Kemiripan Daun Tanaman Semangka dan Gulma

Pada Gambar 2.1. [a], [c], dan [e] merupakan contoh citra daun varietas tanaman semangka; [b] dan [d] menunjukkan contoh citra daun gulma *buffalobur nightshade* (plants.usda.gov dan fireflyforest.net), sedangkan [f] merupakan citra daun tanaman melon.

## 2.3 Citra Warna

Gonzalez dan Woods (2008) menyatakan bahwa warna merupakan hasil persepsi dari cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dan memiliki panjang gelombang antara 400nm sampai dengan 700nm. Suatu model warna adalah model matematis abstrak yang menggambarkan cara agar suatu warna dapat direpresentasikan sebagai baris angka, biasanya dengan nilai-nilai dari tiga atau empat buah warna atau komponen, misalnya RGB (*Red Green Blue*), CMYK (*Cyan Magenta Yellow Key/Black*), HSI (*Hue Satruation Lightness*), atau HSV (*Hue Satruation Value*).

Pada HSV komponen *Hue* menyatakan warna sebenarnya, seperti merah, violet, dan kuning dan digunakan untuk menentukan kemerahan (*redness*) dan kehijauan (*greeness*). *Saturation* merupakan salah satu elemen warna HSV yang mewakili tingkat intensitas warna. Pada nilai tingkat kecerahan (*value*) yang sama nilai *saturation* akan menggambarkan kedekatan suatu warna pada tingkat keabuan. Pada sistem nilai saturasi memiliki rentang antara 0 (minimum) dan 1 atau 100% (maksimum). Dalam ruang warna HSV, untuk merepresentasikan tingkat kecerahan warna digunakan elemen *value*. Pada nilai *value* maksimum, warna yang dihasilkan adalah warna dengan tingkat kecerahan maksimum (warna putih) sedangkan pada *value* minimum dihasilkan warna dengan tingkat kecerahan minimum (warna hitam). Dengan kata lain, pada nilai intensitas komponen *value* yang maksimum dan minimum, intensitas komponen *hue* dan *saturation* tidak berpengaruh.

Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia dan tidak terpengaruh pada intensitas cahaya. Sedangkan warna yang dibentuk model lain seperti RGB merupakan hasil campuran dari warna-warna primer merah, hijau, dan biru berdasarkan komposisi tertentu. Oleh karena itu, diperlukan proses konversi ruang warna RGB ke ruang warna HSV. Persamaan 2.1 sampai dengan Persamaan 2.3 adalah rumus konversi citra dari ruang warna RGB ke ruang warna HSV secara umum (Eldahshan, 2014):

$$V = \frac{\max(R,G,B)}{255} \quad (2.1)$$



$$S = \begin{cases} 0, & \text{jika } V = 0 \\ \frac{\text{maks}(R,G,B) - \min(R,G,B)}{\text{maks}(R,G,B)}, & \text{jika } V \neq 0 \end{cases} \quad (2.2)$$

$$H' = \begin{cases} \frac{60(G-B)}{\text{maks}(R,G,B) - \min(R,G,B)}, & \text{jika } \text{maks}(R,G,B) = R \text{ dan } G \geq B \\ 300 + \frac{60(B-G)}{\text{maks}(R,G,B) - \min(R,G,B)}, & \text{jika } \text{maks}(R,G,B) = R \text{ dan } B \geq G \\ 120 + \frac{60(B-R)}{\text{maks}(R,G,B) - \min(R,G,B)}, & \text{jika } \text{maks}(R,G,B) = G \\ 240 + \frac{60(R-G)}{\text{maks}(R,G,B) - \min(R,G,B)}, & \text{jika } \text{maks}(R,G,B) = B \end{cases}$$

$$H = \frac{H'}{360} \quad (2.3)$$

## 2.4 Segmentasi

Gonzalez dan Woods (2008) menyatakan bahwa segmentasi merupakan membagi citra menjadi beberapa wilayah atau obyek. Beberapa metode segmentasi yang dapat digunakan antara lain deteksi titik, deteksi garis, dan deteksi tepi. Pada penelitian ini akan digunakan segmentasi dengan deteksi tepi menggunakan operator Canny. Tepi adalah perubahan nilai intensitas derajat keabuan yang cepat dalam jarak yang singkat. Sedangkan deteksi tepi pada suatu citra merupakan proses yang menghasilkan tepi-tepi dari objek-objek citra. Suatu titik  $(x, y)$  dikatakan tepi dari suatu citra bila titik tersebut memiliki perbedaan yang tinggi dengan tetangganya. Proses pendeteksian tepi citra digital dapat dilakukan dengan teknik konvolusi menggunakan berbagai macam metode/operator. Operator deteksi tepi merupakan alat yang digunakan untuk memodifikasi nilai derajat keabuan sebuah titik berdasarkan derajat keabuan titik-titik yang ada disekitarnya (konvolusi/operasi ketetanggaan).

### 2.4.1 Deteksi Tepi dengan Operator Canny

Pendekatan deteksi tepi dengan operator Canny berdasarkan tiga tujuan utama, yaitu :

- Optimal detection*,
- Good localization*, titik tepi dapat dilokalisasi dengan baik dengan jarak minimal antara posisi tepi sebenarnya dengan posisi yang terdeteksi, dan
- Single response*, memiliki respon titik tepi yang tunggal dengan mengeliminasi multiple response menjadi satu garis tepi saja.

Algoritma deteksi tepi dengan operator Canny terdiri dari beberapa tahap sebagai berikut :

- Menghaluskan gambar atau menghilangkan *noise* dengan *Gaussian filter*, sehingga diperoleh citra yang dihaluskan  $G(x, y)$ .
- Menemukan gradien citra dengan cara mengkonvolusi setiap titik dalam citra yang telah dihaluskan dengan *mask* turunan pertama terhadap  $x$  dan  $y$  sebagai berikut :

$$P(x, y) = G(x, y) * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad Q(x, y) = G(x, y) * \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad (2.4)$$

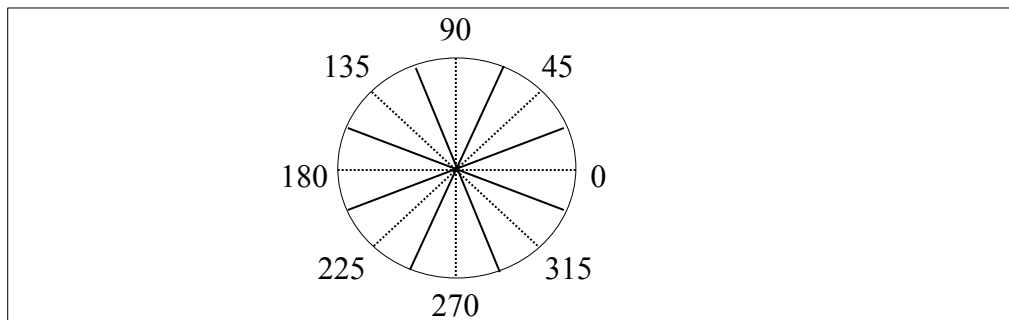
- Menghitung persamaan *magnitude gradient*

$$mag(\nabla f) = \sqrt{P(x, y)^2 + Q(x, y)^2} \quad (2.5)$$

dan arah *gradient*

$$\theta(x, y) = \tan^{-1} \left[ \frac{Q(x, y)}{P(x, y)} \right] \quad (2.6)$$

- Mengaplikasikan *nonmaxima suppression* (menipiskan tepi).



Gambar 2.2 Diagram Arah Gradien dalam Operator Canny

Gambar 2.2 merepresentasikan arah gradien yang digunakan dalam operator Canny. Sudut  $0^\circ$  dan  $180^\circ$  diberi kode 0, sudut  $45^\circ$  dan  $225^\circ$  diberi kode 1, sudut  $90^\circ$  dan  $270^\circ$  diberi kode 2, dan sudut  $135^\circ$  dan  $315^\circ$  diberi kode 3. Penindihan dilakukan dengan ketentuan berdasarkan Gambar 2.2 sebagai berikut:

- Sudut  $0^\circ$ , titik tersebut dianggap sebagai tepi jika intensitasnya lebih besar dari intensitas arah utara dan selatan.
- Sudut  $45^\circ$ , titik tersebut dianggap sebagai tepi jika intensitasnya lebih besar dari intensitas arah barat laut dan tenggara.



- iii) Sudut  $90^0$ , titik tersebut dianggap sebagai tepi jika intensitasnya lebih besar dari intensitas arah barat dan timur.
  - iv) Sudut  $135^0$ , titik tersebut dianggap sebagai tepi jika intensitasnya lebih besar dari intensitas arah timur laut dan barat daya.
  - v) Jika intensitas pada titik tersebut lebih besar dari satu arah dan lebih kecil dari arah lainnya atau sebaliknya, maka titik tersebut tetap dianggap sebagai tepi.
- e. *Threshold* dipilih sedemikian rupa untuk menghilangkan *shadow*. Jika penetapan *threshold* terlalu rendah, maka masih akan ada *shadow*. Sebaliknya, tepi sebenarnya akan dihilangkan jika penetapan *threshold* terlalu tinggi. Algoritma *Canny* berusaha untuk memperbaiki situasi ini dengan menggunakan *hysteresis thresholding*.

## 2.5 Ekstraksi Fitur

Wu, dkk (2007) mengatakan bahwa ekstraksi fitur citra merupakan langkah penting dalam identifikasi pola citra. Ekstraksi fitur harus mempertimbangkan masukan ciri yang membangun struktur informasi untuk klasifikasi. Dalam penelitian ini, fitur yang diekstraksi adalah fitur bentuk, dan tekstur.

### 2.5.1 Fitur Bentuk

Fitur bentuk yang digunakan dalam penelitian ini menggunakan 5 fitur dasar geometris yaitu diameter, panjang fisiologis, lebar fisiologis, area daun, dan *perimeter* daun; dan morfologi digital yang mencakup aspek rasio, *rectangularity*, faktor *narrow*, rasio *perimeter* dari diameter, dan rasio *perimeter* dari panjang fisiologis dan lebar fisiologis.

#### a. Fitur Dasar Geometri

Terdapat 5 fitur dasar geometri ada, yaitu :

##### i. Diameter (D)

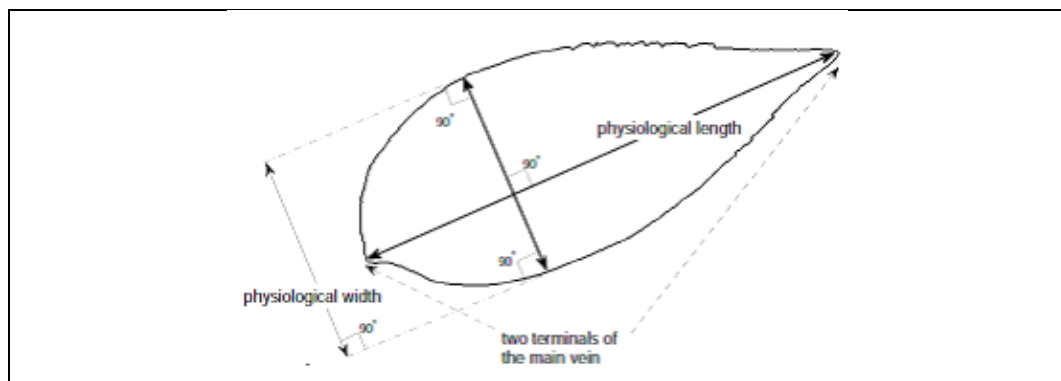
Diameter merupakan jarak terpanjang antara dua titik pada *margin* daun.

ii. Panjang Fisiologis ( $P_f$ )

Panjang fisiologis didefinisikan sebagai jarak antara dua terminal vena daun.

iii. Lebar Fisiologis ( $L_f$ )

Lebar fisiologis didefinisikan sebagai jarak terpanjang antara perpotongan *physiological width* dan dua terminal vena daun yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 Hubungan Panjang Fisiologis dan Lebar Fisiologis

iv. Area Daun ( $A$ )

Nilai dari area daun dievaluasi dengan cara menghitung banyaknya piksel dari citra biner (hitam putih) yang bernilai 1.

v. *Perimeter* Daun ( $P$ )

Perimeter daun adalah jumlah perhitungan piksel yang mengandung *margin* daun.

**b. Fitur Morfologi Digital**

Berdasarkan 5 fitur dasar yang diperkenalkan sebelumnya, dapat didefinisikan ciri-ciri morfologi digital yang digunakan untuk daun yaitu :

i. Aspek Rasio (*Slimness*)

Aspek rasio didefinisikan sebagai rasio panjang fisiologis ( $P_f$ ) dengan lebar fisiologis ( $L_f$ ) yang dapat dituliskan dengan

$$A_r = \frac{P_f}{L_f} \quad (2.7)$$



ii. Faktor *Form (Roundness)*

Faktor *form* digunakan untuk mendeskripsikan kemiripan antara daun dan lingkaran yang dilakukan untuk mengetahui seberapa bundar bentuk daun itu. Faktor *form* dihitung dengan persamaan:

$$F = \frac{4\pi A}{P^2} \quad (2.8)$$

iii. *Rectangularity*

*Rectangularity* menggambarkan kesamaan antara daun dan persegi panjang. *Rectangularity* dapat dinotasikan sebagai berikut:

$$Rect = \frac{P_f L_f}{A} \quad (2.9)$$

iv. Faktor *Narrow*

Faktor *Narrow* didefinisikan sebagai rasio diameter D dan panjang fisiologis. Persamaan faktor *narrow* dinotasikan sebagai berikut:

$$N = \frac{D}{P_f} \quad (2.10)$$

v. Rasio *Perimeter* dari Diameter

Rasio *perimeter* dari diameter merepresentasikan rasio dari perimeter daun dan diameter daun, dirumuskan dengan:

$$P_D = \frac{P}{D} \quad (2.11)$$

vi. Rasio *Perimeter* dari Panjang Fisiologis dan Lebar Fisiologis

Fitur ini mendefinisikan rasio dari *perimeter* daun dan jumlah dari panjang fisiologis dan lebar fisiologis, sehingga didapatkan persamaan:

$$P_{PLF} = \frac{P}{P_f + L_f} \quad (2.12)$$

### 2.5.2 Fitur Tekstur

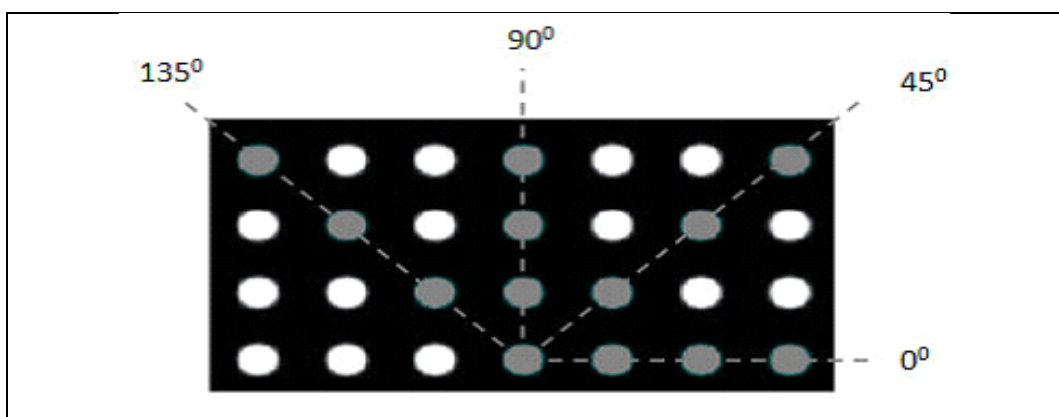
Fitur tekstur adalah sekumpulan standar pengukuran yang dihitung dalam pengolahan citra yang dimaksudkan untuk menghitung tekstur dari citra daun. Tekstur citra daun memberikan informasi mengenai tata ruang warna atau

intensitas dalam citra daun atau wilayah yang dipilih dari citra daun. Fitur tekstur diekstrak menggunakan *Gray Level Co-occurrence Matrices* (GLCM) yang sangat berguna untuk memperoleh informasi berharga tentang frekuensi relatif dengan dua piksel yang dipisahkan oleh jarak piksel dan sudut tertentu. Satu dengan intensitas  $i$  dan yang lainnya dengan intensitas  $j$  (Kadir dkk, 2011). Fitur *co-occurrence* GLCM ini mencari nilai *contrast*, *correlation*, *energy*, dan *homogeneity*. Matriks *co-occurrence* GLCM menghitung piksel *co-occurrence* dengan nilai *gray* dan pada jarak tertentu. Fitur *Co-occurrence* yang diperhatikan pada *Gray Level Co-occurrence Matrices* ditunjukkan pada Tabel 1.

Tabel 1. *Co-ocurrence Features*

No.	Nama Fitur	Kalkulasi
1.	<i>Contrast</i>	$Contrast = \sum_{i,j}  i - j ^2 P_{\theta}(i, j)$
2.	<i>Correlation</i>	$Correlation = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) P_{\theta}(i, j)}{\sigma_i \sigma_j}$
3.	<i>Energy</i>	$Energy = \sum_{i,j} P_{\theta}(i, j)^2$
4.	<i>Homogeneity</i>	$Homogeneity = \sum_{i,j} \frac{P_{\theta}(i, j)}{1 +  i - j }$

Sudut yang digunakan untuk memperoleh fitur *contrast*, *correlation*, *energy* dan *homogeneity* adalah  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$  (seperti Gambar 2.3) karena sudut  $0^\circ = 180^\circ$ ,  $45^\circ = 225^\circ$ ,  $90^\circ = 270^\circ$ , dan  $135^\circ = 315^\circ$ .



Gambar 2.4 Sudut dalam GLCM



## 2.6 Klasifikasi dengan *Support Vector Machine* (SVM)

Klasifikasi adalah proses menemukan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep. Model yang diturunkan berdasarkan analisis dari set data pelatihan (data yang label kelasnya diketahui) yang kemudian digunakan untuk memprediksi label kelas yang tidak diketahui. Banyak metode klasifikasi telah diperkenalkan peneliti seperti *machine learning*, *pattern recognition*, dan statistik (Han dkk, 2012).

Pada tahun 1992 *Support Vector Machine* (SVM) diperkenalkan oleh Vladimir Vapnik, Bernhard Boser dan Isabelle Guyon. Sebelum tahun 1992 tepatnya tahun 1960-an dasar untuk SVM sudah ada (termasuk karya awal oleh Vapnik dan Alexei Chervonenkis pada teori belajar statistik). *Support Vector Machine* (SVM) adalah klasifikasi jenis *supervised* karena ketika proses *training* diperlukan target pembelajaran tertentu. SVM merupakan algoritma yang bekerja untuk memecahkan masalah klasifikasi dengan mencari *maximum marginal hyperplane* (MMH). Konsep SVM secara sederhana dapat dinyatakan sebagai usaha menemukan fungsi pemisah (*classifier/hyperplane*) yang optimal yang bisa memisahkan dua set data dari dua kelas yang berbeda.

Misalkan diberikan 2 atribut input  $A_1$  dan  $A_2$ , himpunan  $W = \{w_1, w_2, \dots, w_d\}$ ;  $W$  merupakan bobot/*weight*,  $d$  adalah jumlah atribut, dan tupel *training*  $X = (x_1, x_2)$  dimana  $x_1$  dan  $x_2$  adalah nilai-nilai atribut  $A_1$  dan  $A_2$ , maka fungsi *hyperplane* dapat dinotasikan sebagai berikut:

$$f(x) = W \cdot X + b \quad (2.13)$$

dimana  $W, X \in \mathbb{R}^d$  ( $d$  adalah jumlah atribut) dan  $b$  adalah bias yang berupa skalar (Hamel, 2009).

*Hyperplane* yang terletak diantara dua set objek dari kelas positif ( $y_1 = +1$ ) dan kelas negatif ( $y_2 = -1$ ) dapat ditulis sebagai berikut:

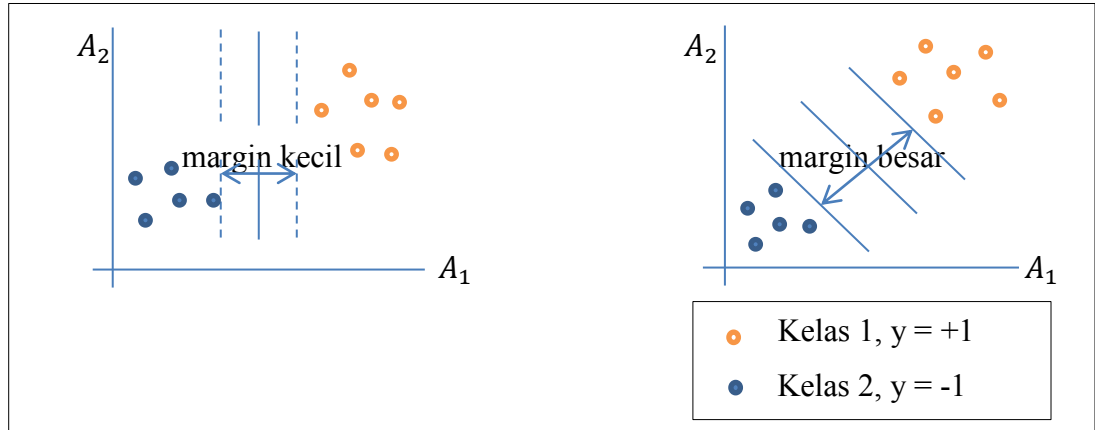
$$H_1: x_i w + b \geq 1 \text{ untuk } y_1 = +1 \quad (2.14)$$

$$H_2: x_i w + b \leq -1 \text{ untuk } y_2 = -1 \quad (2.15)$$

Penggabungan dari persamaan (2.14) dan (2.15) menghasilkan pertidaksamaan:

$$y_i(x_i w + b) \geq 1, \text{ untuk } \forall i = 1, 2, \dots, n \quad (2.16)$$

dengan  $n$  merupakan jumlah data training.



Gambar 2.5 Margin Kecil dan Margin Besar

Margin antara dua kelas dapat dihitung dengan mencari jarak antara kedua *hyperplane*  $H_1$  atau  $H_2$ . Setiap tupel pelatihan yang jatuh pada *hyperplane*  $H_1$  atau  $H_2$  yang memenuhi persamaan (2.14) disebut *support vector*. Jarak terdekat suatu titik di bidang  $H_1$  terhadap pusat dapat dihitung dengan meminimalkan  $x^T x$  dengan memperhatikan kendala  $x_i w + b \geq 1$ . Dengan menggunakan bentuk fungsi *Lagrange* dan turunan pertamanya diperoleh:

$$\begin{aligned} \min x^T x - \lambda(w^T x + b - 1) \\ \frac{d}{dx}(x^T x - \lambda(w^T x + b - 1)) &= 0 \\ \Rightarrow 2x - \lambda w &= 0 \\ \Rightarrow x &= \frac{\lambda}{2} w \end{aligned}$$

dengan mensubstitusikan  $x$  ke bidang  $H_1$ :  $x_i w + b = 1$  diperoleh:

$$\begin{aligned} \frac{\lambda}{2} w^T w + b &= 1 \\ \Rightarrow \lambda &= \frac{2(1-b)}{w^T w} \end{aligned}$$

sehingga dengan mensubstitusikan kembali  $\lambda$  pada  $x$  diperoleh:

$$\begin{aligned} x &= \frac{(1-b)}{w^T w} w \\ x^T x &= \frac{(1-b)^2}{(w^T w)^2} w^T w = \frac{(1-b)^2}{w^T w} \end{aligned}$$

maka jarak  $H_1$  ke pusat adalah:

$$\|x\| = \sqrt{x^T x} = \sqrt{\frac{(1-b)^2}{w^T w}} = \frac{(1-b)}{\|w\|}$$

Untuk mencari jarak terdekat suatu titik pada bidang  $H_2$  terhadap titik pusat dapat dihitung dengan meminimalkan  $x^T x$  dengan memperhatikan kendala  $x_i w + b \leq -1$ . Dengan menggunakan bentuk fungsi *Lagrange* dan turunan pertamanya diperoleh:

$$\min x^T x + \lambda(w^T x + b + 1)$$

$$\frac{d}{dx}(x^T x + \lambda(w^T x + b + 1)) = 0$$

$$\Rightarrow 2x + \lambda w = 0$$

$$\Rightarrow x = -\frac{\lambda}{2} w$$

dengan mensubstitusikan  $x$  ke bidang  $H_2: x_i w + b = -1$  diperoleh:

$$-\frac{\lambda}{2} w^T w + b = -1$$

$$\Rightarrow \lambda = \frac{2(b+1)}{w^T w}$$

Sehingga dengan mensubstitusikan kembali  $\lambda$  pada  $x$  diperoleh:

$$x = \frac{(-b-1)}{w^T w} w$$

$$x^T x = \frac{(-b-1)^2}{(w^T w)^2} w^T w = \frac{(-b-1)^2}{w^T w}$$

maka jarak  $H_1$  ke pusat adalah :

$$\|x\| = \sqrt{x^T x} = \sqrt{\frac{(-b-1)^2}{w^T w}} = \frac{(-b-1)}{\|w\|}$$

margin maksimum didapatkan dengan memaksimalkan jarak antara  $H_1$  dan  $H_2$  :

$$\left| \frac{(1-b)}{\|w\|} - \frac{(-b-1)}{\|w\|} \right| = \frac{2}{\|w\|}$$

Oleh karena memaksimalkan  $\frac{1}{\|w\|}$  sama dengan meminimumkan  $\|w\|^2$

dan untuk menyederhanakan penyelesaian ditambahkan faktor  $\frac{1}{2}$ . Dengan demikian, model persamaannya menjadi:

$$\text{Min } \frac{1}{2} \|w\|^2$$

$y_i(x_i w + b) \geq 1$ , untuk  $\forall i = 1, 2, \dots, n$  ( $n$  merupakan jumlah data training).



Untuk menyelesaikan persamaan tersebut, secara komputasi lebih sulit dan perlu waktu lebih panjang. Untuk itu diperkenalkan pengali Lagrangian  $\alpha_i$ , dengan  $i = 1, 2, \dots, n$ . Sehingga model masalah sebelumnya dapat ditulis:

$$\begin{aligned} \text{Min } L_p &= \frac{1}{2} \|w\|^2 + \alpha_i (0 - \sum_{i=1}^n [y_i(x_i \cdot w + b) - 1]) \\ L_p &= \frac{1}{2} \|w\|^2 + 0 - \alpha_i (\sum_{i=1}^n [y_i(x_i \cdot w + b) - 1]) \\ L_p &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1] \\ L_p &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^n \alpha_i \quad (2.17) \end{aligned}$$

Solusi dapat diperoleh dengan meminimalkan  $L_p$  terhadap variabel  $w, b$  (*primal variable*):

$$\begin{aligned} \frac{\partial L_p}{\partial b} &= 0 \\ \sum_{i=1}^n \partial_i y_i &= 0 \quad (2.18) \\ \text{dan} \\ \frac{\partial L_p}{\partial w} &= 0 \\ w - \sum_{i=1}^n \alpha_i y_i x_i &= 0 \\ w &= \sum_{i=1}^n \alpha_i y_i x_i \quad (2.19) \end{aligned}$$

Dan memaksimalkan  $L_D$  terhadap variabel  $\alpha_i$  (*dual variables*). Untuk mendapatkan problem dual ( $L_D$ ), substitusikan persamaan (2.18) dan (2.19) ke (2.16):

$$\begin{aligned} \text{Maks } L_D &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1] \\ \|w\|^2 &= (w \cdot w) \\ &= (\sum_{i=1}^n \alpha_i y_i x_i \cdot \sum_{j=1}^n \alpha_j y_j x_j) \\ &= \underbrace{\sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j}_A \quad (2.20) \end{aligned}$$

dan

$$\begin{aligned} \sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1] &= \sum_{i=1}^n \alpha_i y_i x_i w + \sum_{i=1}^n \alpha_i y_i b - \sum_{i=1}^n \alpha_i \\ &= \sum_{i=1}^n \alpha_i y_i x_i \sum_{j=1}^n \alpha_j y_j x_j + 0 - \sum_{i=1}^n \alpha_i \\ &= \underbrace{\sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j}_A - \underbrace{\sum_{i=1}^n \alpha_i}_B \quad (2.21) \end{aligned}$$

sehingga diperoleh

$$\begin{aligned}
L_D &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1] \\
&= \frac{1}{2} A - (A - B) \\
&= \frac{1}{2} A - A + B \\
&= -\frac{1}{2} A + B \\
&= B - \frac{1}{2} A \\
&= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j \quad (2.22) \\
\sum_{i=1}^n \alpha_i y_i &= 0, \text{ dan } 0 \leq \alpha_i, i = 1, 2, \dots, n
\end{aligned}$$

Dengan demikian, dapat diperoleh nilai  $\alpha_i$  yang akan digunakan untuk menentukan  $w$  dan  $b$  dengan rumus:

$$\begin{aligned}
w &= \sum_{i=1}^n \alpha_i y_i x_i \\
b &= y_i - w x_i \quad (2.23)
\end{aligned}$$

Terdapat nilai  $\alpha_i$  untuk setiap data pelatihan. Data pelatihan yang memiliki nilai  $\alpha_i > 0$  adalah *support vector*. Dengan demikian fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*.

Berdasarkan formula *lagrangian*, MMH dapat didefinisikan sebagai *boundary decision*:

$$f(x_d) = \sum_{i=1}^l \alpha_i y_i x_i + b \quad (2.24)$$

dimana  $y_i$  adalah label kelas *support vector*  $X_i$ ,  $X_z$  adalah data uji,  $\alpha_i$  dan  $b$  adalah parameter numerik yang ditentukan secara otomatis oleh optimasi algoritma SVM dan  $l$  adalah jumlah *support vector*.

Dalam klasifikasi kadang-kadang dijumpai bidang pemisah yang tidak bisa diambil dengan linier sehingga diperlukan penyelesaian khusus untuk permasalahan ini. Untuk data-data yang tidak dapat dipisahkan secara linier tersebut ditambahkan variabel *slack*  $\xi_i \geq 0$  ke pertidaksamaan (2.14) sehingga kendala dan fungsi tujuan menjadi:

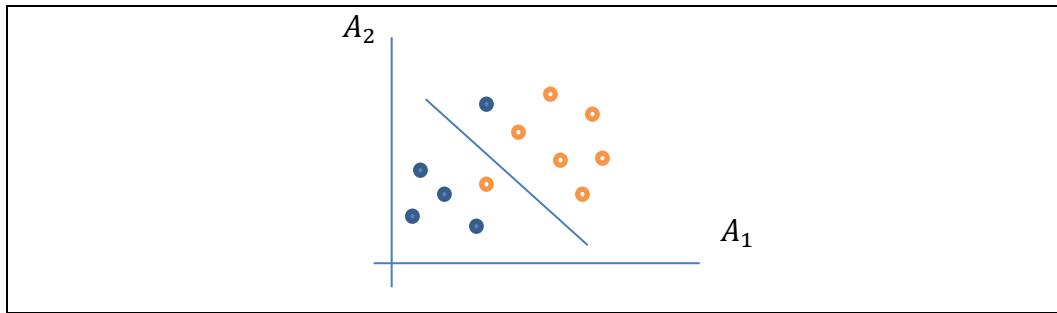
$$y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0, \text{ untuk } \forall_i = 1, 2, \dots, n \quad (2.25)$$

dengan  $n$  merupakan jumlah data *training*.

$$\text{Min } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (2.26)$$

dengan  $y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0, \xi_i \geq 0, \text{ untuk } \forall_i$ .

Dengan meminimalkan  $\|w\|^2$  (Persamaan 2.26), ini juga berarti memaksimalkan margin antara dua kelas. Hal ini dilakukan untuk mengurangi jumlah kesalahan klasifikasi (missclassification error) yang dinyatakan dengan adanya variabel *slack*  $\xi_i$ . Penggunaan variabel *slack* ini adalah untuk mengatasi kasus ketidaklayakan (*infeasibility*) dari pembatas (*constraints*) dengan cara memberi pinalti untuk data yang tidak memenuhi pembatas tersebut. Pinalti ini di notasikan dengan C (Vijayakumar dan Wu, 1999).



Gambar 2.6 Sebaran data yang terpisah secara nonlinier

Selanjutnya, bentuk *primal problem* pada Persamaan 2.17 berubah menjadi:

$$\begin{aligned} \text{Min } L_p = & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i (x_i \cdot w + b) - 1 + \xi_i\} \\ & - \sum_{i=1}^n q_i \xi_i \end{aligned} \quad (2.27)$$

Dengan cara yang sama dengan penurunan persamaan dual problem pada data linier, maka persamaan dual problem untuk data nonlinier adalah sebagai berikut:

$$\begin{aligned} \text{Min } L_p = & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i (x_i \cdot w + b) - 1 + \xi_i\} - \sum_{i=1}^n q_i \xi_i \\ \|w\|^2 = & (w \cdot w) \\ = & \left( \sum_{i=1}^n \alpha_i y_i x_i \cdot \sum_{j=1}^n \alpha_j y_j x_j \right) \\ = & \sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j \end{aligned}$$

dan

$$\begin{aligned} & C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i (x_i \cdot w + b) - 1 + \xi_i\} - \sum_{i=1}^n q_i \xi_i \\ = & C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i x_i w + \sum_{i=1}^n \alpha_i y_i x_i b - \sum_{i=1}^n \alpha_i + \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i \\ = & C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i x_i \sum_{j=1}^n \alpha_j y_j x_j + 0 - \sum_{i=1}^n \alpha_i + \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i \end{aligned}$$



$$= C \sum_{i=1}^n \xi_i - \sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j - \sum_{i=1}^n \alpha_i$$

sehingga diperoleh :

$$L_p(w, x, \xi_i) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i (x_i \cdot w + b) - 1 + \xi_i\} - \sum_{i=1}^n q_i \xi_i \quad (2.28)$$

untuk meminimumkan  $L_p(w, x, \xi_i)$  digunakan  $\frac{\partial L_p}{\partial b} = 0$ ,  $\frac{\partial L_p}{\partial w} = 0$ , dan  $\frac{\partial L_p}{\partial \xi_i} = 0$

dengan  $i = 1, 2, \dots, n$  ( $n$  merupakan jumlah data training). Maka

$$\begin{aligned} \frac{\partial L_p}{\partial b} &= \sum_{i=1}^n \alpha_i y_i = 0 \\ \frac{\partial L_p}{\partial w} &= w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \end{aligned}$$

dengan  $w = \sum_{i=1}^n \alpha_i y_i x_i$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - q_i = 0$$

Akibatnya,  $C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i = 0$

Dengan memanfaatkan turunan parsial  $L_p$  terhadap  $w, b$  dan  $\xi_i$ , masalah dual menjadi:

$$\begin{aligned} Maks L_D &= \frac{1}{2} (\sum_{i=1}^n \alpha_i y_i x_i) (\sum_{j=1}^n \alpha_j y_j x_j) + C \sum_{i=1}^n \xi_i \\ &\quad - \sum_{i=1}^n \alpha_i y_i ((\sum_{i=1}^n \alpha_i y_i x_i) \cdot x_i + b) + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j (x_i \cdot x_j) - \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j (x_i \cdot x_j) \\ &\quad - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j (x_i \cdot x_j) \end{aligned} \quad (2.29)$$

dengan kendala

$$\sum_{i=1}^n \alpha_i y_i = 0$$

karena

$$\frac{\partial L_D}{\partial \xi_i} = C - \alpha_i - q_i = 0$$

dan semua nilai  $q_i$  tidak diperhitungkan pada fungsi tujuan, maka kendalanya menjadi

$$0 \leq \alpha_i \leq C, \text{ untuk } i = 1, 2, \dots, n$$

Bias yang sebelumnya telah didefinisikan sebagai  $b$ , dapat didefinisikan secara implisit sebagai atribut tambahan. Sehingga data *input*

$X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$  menjadi  $X_j^T = [x_{i1}, x_{i2}, \dots, x_{im}\lambda]$  dan vektor bobot  $w$  menjadi  $w^T = w_1, w_2, \dots, w_m, b/\lambda$  dengan  $\lambda$  adalah konstanta.

bentuk permasalahan menjadi meminimumkan:

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - y_i)(x_i^T \cdot w^T)(-\xi_i) + \sum_{i=1}^n q_i(-\xi_i)$$

untuk meminimumkan  $L_p$  digunakan  $\frac{\partial L_p}{\partial w^T} = 0$ , dan  $\frac{\partial L_p}{\partial \xi_i} = 0$  dengan  $i = 1, 2, \dots, n$

dengan  $n$  merupakan jumlah data *training*. Maka

$$\frac{\partial L_p}{\partial w^T} = w^T - \sum_{i=1}^n \alpha_i y_i X_i = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i y_i X_i$$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - q_i = 0$$

Bentuk dual dari masalah itu adalah:

$$\begin{aligned} L_D &= \frac{1}{2} (\sum_{i=1}^n \alpha_i y_i X_i) (\sum_{j=1}^n \alpha_j y_j X_j^T) + C \sum_{i=1}^n \xi_i \\ &\quad + \sum_{i=1}^n \alpha_i (1 - y_i (\sum_{j=1}^n \alpha_j y_j X_j^T \cdot X_i^T) - \xi_i) + \sum_{i=1}^n q_i (-\xi_i) \\ &= \frac{1}{2} \sum_{i=1}^n \alpha_i y_i X_i \sum_{j=1}^n \alpha_j y_j X_j^T + \sum_{i=1}^n \alpha_i (y_i (\sum_{j=1}^n \alpha_j y_j X_j^T \cdot X_i^T)) \\ &\quad + \sum_{i=1}^n \alpha_i + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \alpha_i y_i X_i \sum_{j=1}^n \alpha_j y_j X_j^T \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (X_i \cdot X_j^T) \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(X_i \cdot X_j^T) \end{aligned} \tag{2.30}$$

Dengan kendala

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, n$$

### Metode Kernel

Secara umum, kasus klasifikasi di dunia nyata adalah kasus yang tidak linier sehingga digunakan metode kernel untuk mengatasi masalah tersebut. Dengan menggunakan metode kernel suatu data  $x$  di *input space* dipetakan ke *feature space*  $F$  dengan dimensi yang lebih tinggi melalui *map*  $\varphi$  ( $\varphi: x \rightarrow \varphi(x)$ ). Oleh karena itu,  $x$  di *input space* menjadi  $\varphi(x)$  di *feature space*. Sering kali fungsi  $\varphi(x)$  tidak tersedia atau tidak dapat dihitung. Tetapi *dot product* dari dua vektor dapat dihitung, baik dalam *input space* maupun di *feature space*.

$$K(X_i \cdot X_j^T) = \varphi(X_i) \cdot \varphi(X_j) \tag{2.31}$$

Fungsi kernel yang umum digunakan adalah sebagai berikut:

$$\text{Kernel Derajat Polinomial } h : K(X_i \cdot X_j^T) = (X_i \cdot X_j^T + 1)^h \quad (2.32)$$

$$\text{Kernel Fungsi Gaussian Radial Basis : } K(X_i \cdot X_j^T) = \frac{e^{-\|x_i - x_j\|^2}}{2\sigma^2} \quad (2.33)$$

$$\text{Kernel Sigmoid : } K(X_i \cdot X_j^T) = \tanh(X_i \cdot X_j^T - \delta) \quad (2.34)$$

Berdasarkan formula *lagrangian*, MMH dapat didefinisikan sebagai *boundary decision*:

$$f_i = \text{sign} \left( \sum_{i=1}^l \alpha_i y_i \varphi(X_i) \cdot \varphi(X_z) + b \right) \quad (2.35)$$

Keterangan:

$\varphi(X_i) \cdot \varphi(X_z)$  = *inner product* antara data *support vector* dengan data uji.

sign = fungsi *signum* yaitu nilai yang lebih besar dari 0 dilabelkan +1 dan semua nilai yang lebih kecil dari 0 dilabelkan -1.

$X_z$  =  $(z_{11}, z_{12}, \dots, z_{1k})$ , k adalah jumlah data uji.

## 2.7 Penelitian-Penelitian yang Relevan

Kalyoncu dan Toygar, (2014) mengklasifikasikan daun dari segi geometri. Segmentasi dilakukan untuk menghapus noise yang mengganggu proses geometri digital yang ada. Setelah itu mengekstraksi fitur yang dilakukan dengan mencari momen invarian, konveksitas, rasio perimeter, matriks jarak multiskala, jarak rata-rata margin, dan statistik margin. Metode klasifikasi yang digunakan adalah *Linier Discriminant Classification* (LDC) karena metode ini memprioritaskan fitur tergantung pada jenis daun dan memiliki kompleksitas yang lebih rendah atau sama dibandingkan dengan metode klasifikasi yang lain.

Wang, dkk (2008) menggunakan metode *watershed* yang dikombinasikan dengan prasegmentasi dan operasi morfologi yang diaplikasikan untuk mensegmentasi gambar daun dengan latar belakang yang rumit berdasarkan informasi prior daun. Klasifikasi daun dilakukan dengan menggunakan klasifikasi *moving center hypersphere* (MCH).



Metode *Support Vector Machine* (SVM) untuk klasifikasi daun tanaman dan pohon dilakukan oleh Narayan dan Subbarayan, (2014) dengan mengoptimalkan seleksi subset fitur menggunakan *Genetic Algorithm* (GA). Fitur yang digunakan adalah area daun, standar deviasi, *boundary*, jumlah riak dan jumlah piksel riak. Metode yang diajukan ini terbukti memiliki tingkat akurasi yang lebih tinggi dibandingkan *K-Nearest Neighbors* (KNN).

Arribas, dkk. (2011) mengklasifikasikan daun dari tanaman bunga matahari dengan menggunakan visi komputer dan *neural network*. Tujuan utama dari penelitian ini untuk membedakan antara bunga matahari dan gulma berdasarkan ruang warna menggunakan *Generalized Softmax Perceptron* (GSP) arsitektur jaringan saraf yang di-*training* dengan bantuan algoritma *Posterior Probability Model Selection* (PPMS).

## **BAB 3**

### **METODOLOGI PENELITIAN**

#### **3.1 Tahapan Penelitian**

##### **3.1.1 Akuisisi**

Pada tahap ini dilakukan proses pengambilan data. Data diambil dengan format citra digital yang berupa jpg dengan menggunakan kamera. Objek yang digunakan adalah daun dari tanaman semangka dan gulma. Data *training* dalam penelitian ini adalah citra tunggal daun semangka dengan berbagai kondisi, seperti silau karena terkena sinar matahari, bibit, usia tanaman 15 hari dan 40 hari, serta kondisi daun yang terpotong. Data *testing* yang digunakan adalah sebuah citra *multi* daun yang diambil di sawah yang terdiri dari daun semangka dan gulma. Data *training* dan data *testing* yang digunakan dalam penelitian ini adalah citra daun dengan *background* secarik kertas putih.

##### **3.1.2 Preprocessing Citra**

Beberapa langkah yang harus dilakukan pada tahap *preprocessing* citra adalah sebagai berikut:

###### **a. Cropping Citra**

Pada saat pengambilan citra digital terdapat bagian yang tidak diinginkan, untuk menghilangkan bagian tersebut maka dilakukan *cropping* citra.

###### **b. Resize Citra**

*Resize* citra adalah pengubahan ukuran piksel citra dari ukuran besar menjadi ukuran yang lebih kecil untuk memudahkan proses pendeteksian

###### **c. Pemisahan *Background* dan *Foreground* Citra serta Menghilangkan Objek yang Tidak Diinginkan**

Sebelum melakukan segmentasi citra, hal yang perlu dilakukan adalah memisahkan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan pada citra yang telah diambil. Pada saat pengambilan citra daun di lapangan, kendala yang sering terjadi adalah batang, buah, bunga tanaman ikut termuat dalam pengambilan citra digital dan juga terdapat bayangan pada citra

yang diambil. Oleh karena itu, diperlukan pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan dengan cara mengubah citra RGB menjadi citra HSV. Kemudian mengambil komponen H dan S. Dan jika terdapat area dari suatu objek dalam citra berukuran lebih kecil dari 2000 piksel, maka objek tersebut dihilangkan.

#### **d. Segmentasi Citra**

Segmentasi citra yang akan digunakan pada penelitian ini adalah deteksi tepi menggunakan operator *Canny*.

#### **3.1.3 Ekstraksi Fitur**

Citra daun yang telah melalui *preprocessing* merupakan data masukan pada ekstraksi fitur karena lebih memudahkan dalam proses ekstraksi fitur bentuk dan tekstur. Fitur bentuk yang diambil adalah *rectangularity*, faktor *narrow*, rasio *perimeter* dari diameter, dan rasio *perimeter* dari panjang fisiologis dan lebar fisiologis. Fitur tekstur diekstraksi dengan menggunakan *Gray Level Co-occurrence Matrices* (GLCM).

#### **3.1.4 Klasifikasi dengan Metode SVM**

Dari hasil ekstraksi fitur yang dilakukan dapat digunakan untuk mengklasifikasikan tanaman tersebut ke dalam jenis tanaman semangka atau gulma dengan menggunakan metode *Support Vector Machine* (SVM). Data yang digunakan untuk *training* merupakan data citra tunggal sebanyak 189 data citra yang terdiri dari 156 data citra semangka dan 33 data citra gulma. Sedangkan data uji menggunakan citra multi daun yang terdiri atas daun tanaman semangka dan gulma sebanyak 19 data citra.

#### **3.1.5 Uji Coba Akurasi Hasil Klasifikasi**

Pada tahap ini dilakukan uji coba akurasi hasil dengan mencocokkan data hasil prediksi klasifikasi menggunakan *Support Vector Machine* (SVM) dengan data asli.

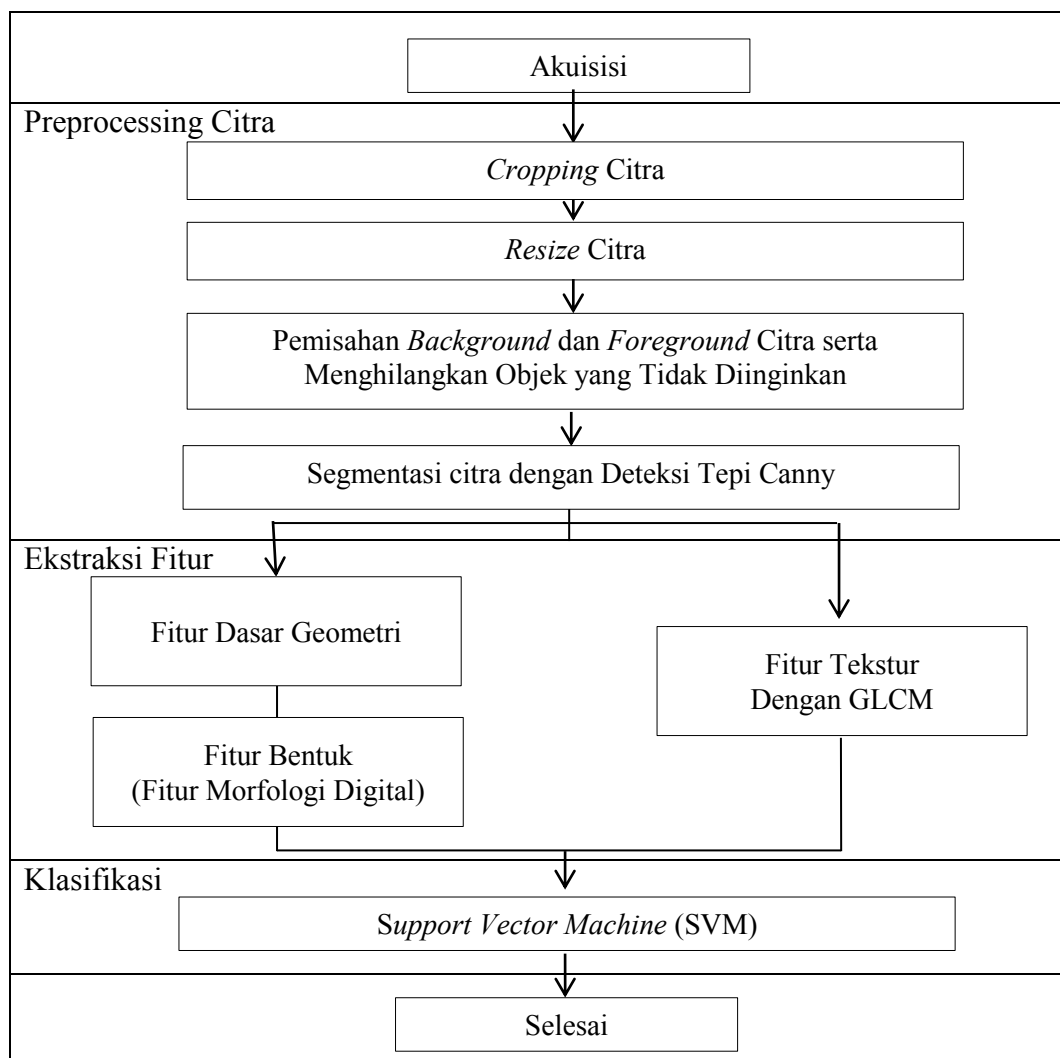


### 3.1.6 Penyusunan Tesis

Setelah mendapatkan hasil akurasi dan mendapatkan persentase dari jumlah daun tanaman semangka dan gulma, maka dapat dilakukan penyusunan tesis sebagai dokumentasi dari penelitian.

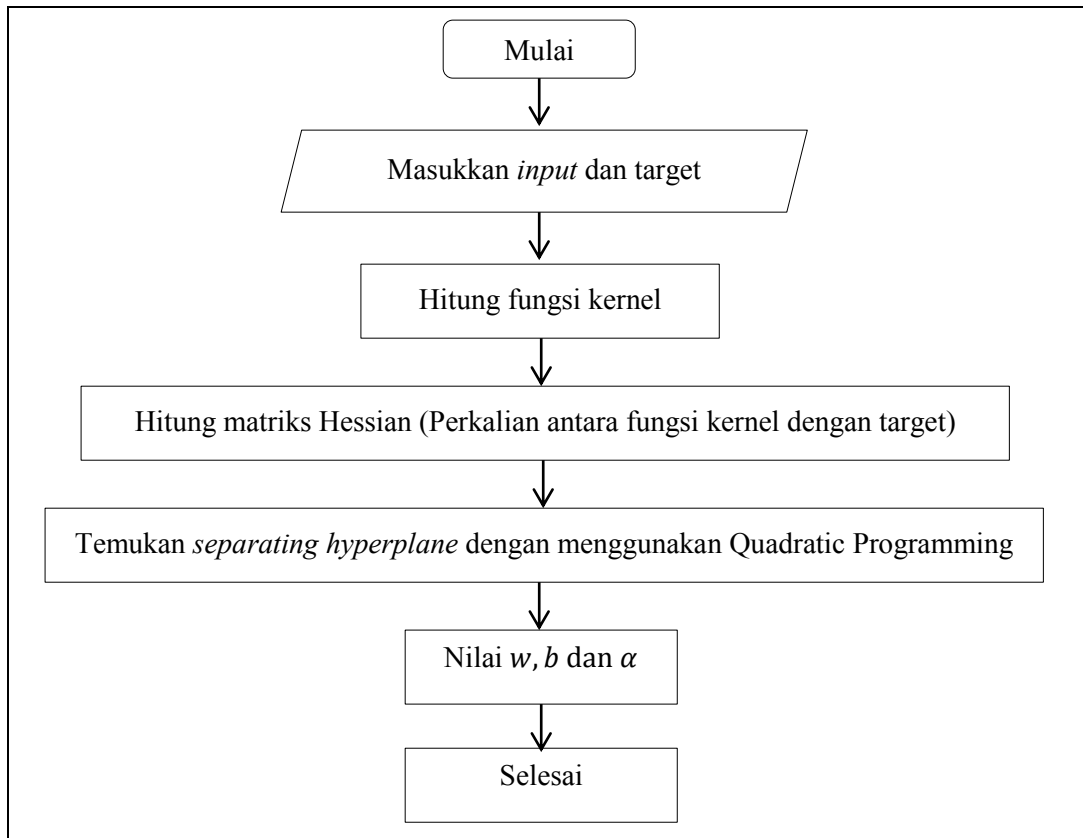
### 3.2 Diagram Proses Penelitian

Diagram alir sistem digunakan untuk menggambarkan langkah kerja sistem yang akan dibuat dan juga akan digunakan oleh peneliti untuk menentukan langkah-langkah berikutnya.



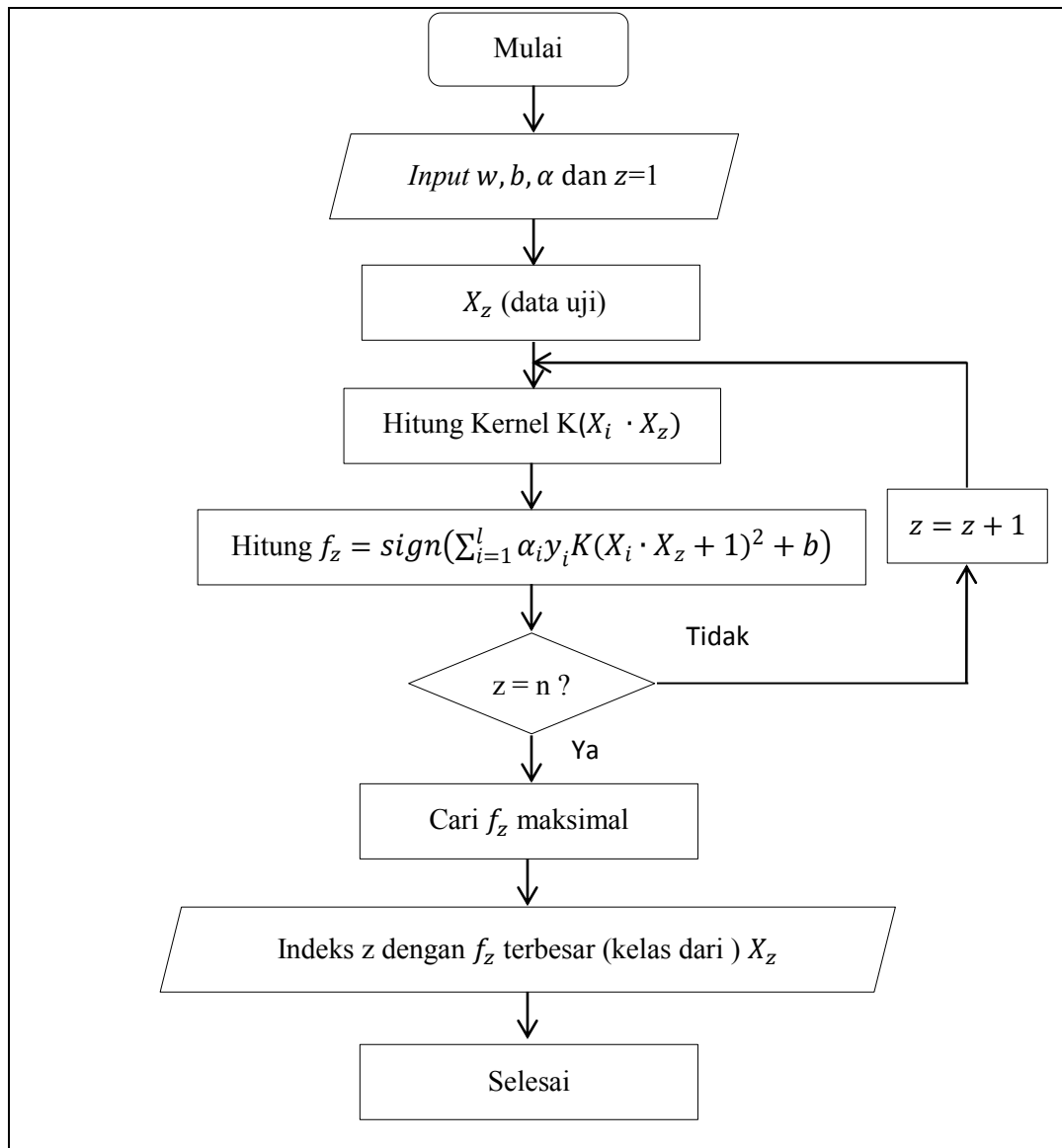
Gambar 3.1. Rancangan Klasifikasi Daun

Untuk metode *Support Vector Machine* (SVM), proses *training* dilakukan untuk memperoleh nilai bobot ( $w$ ),  $\alpha$  dan bias ( $b$ ). Proses *training* selengkapnya dapat dilihat pada diagram alir sebagai berikut:



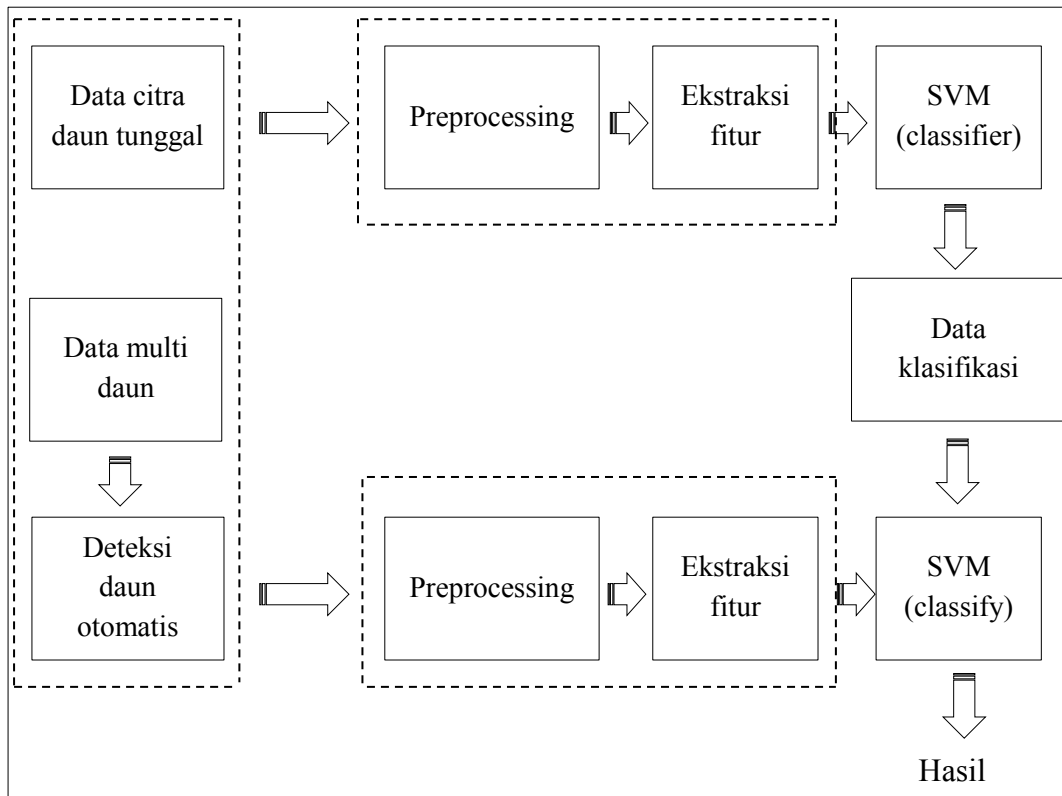
Gambar 3.2 Diagram Alir Proses *Training* SVM

Nilai  $w, b$  dan  $\alpha$  yang kita peroleh akan digunakan pada proses *testing*. Diagram alir proses *testing* ditunjukkan pada gambar berikut:



Gambar 3.3 Diagram Alir Proses Uji SVM

Blok diagram desain sistem yang digunakan pada penelitian ini disajikan pada Gambar 3.4.



Gambar 3.4 Blok Diagram Desain Sistem



## BAB 4

### ANALISIS DAN PEMBAHASAN

Pada bab ini dibahas mengenai penerapan algoritma *Support Vector Machine* (SVM) pada data citra tunggal daun sehingga diperoleh nilai bobot, nilai bias dan nilai *alpha* yang kemudian diterapkan pada data uji citra multi daun, selanjutnya dicari akurasi dari algoritma tersebut.

#### 4.1 Preprocessing

Sebelum melakukan pemrosesan citra digital, *file* citra digital tersebut harus dibaca terlebih dahulu. Perintah untuk membaca *file* citra digital adalah :

```
I = imread('IMG_0164.JPG');
```

Salah satu contoh hasil yang diperoleh dari perintah tersebut ditampilkan pada Gambar 4.1.



Gambar 4.1 Citra Asli

Setelah *file* dibaca dan disimpan dengan variabel “I”, maka informasi dari *file* tersebut dapat terlihat. Misalnya, untuk melihat informasi panjang dan lebar sebuah citra dapat digunakan perintah sebagai berikut:

```
[panjang lebar] = size(I);
```

Dengan perintah tersebut, maka diperoleh informasi panjang = 5184 dan lebar = 3456. *File* citra yang telah dibaca dapat direpresentasikan dalam bentuk matriks, namun karena ukuran *file* citra digital pada Gambar 4.1 adalah  $5184 \times 3456$  sehingga terlalu besar untuk menampilkan semuanya.

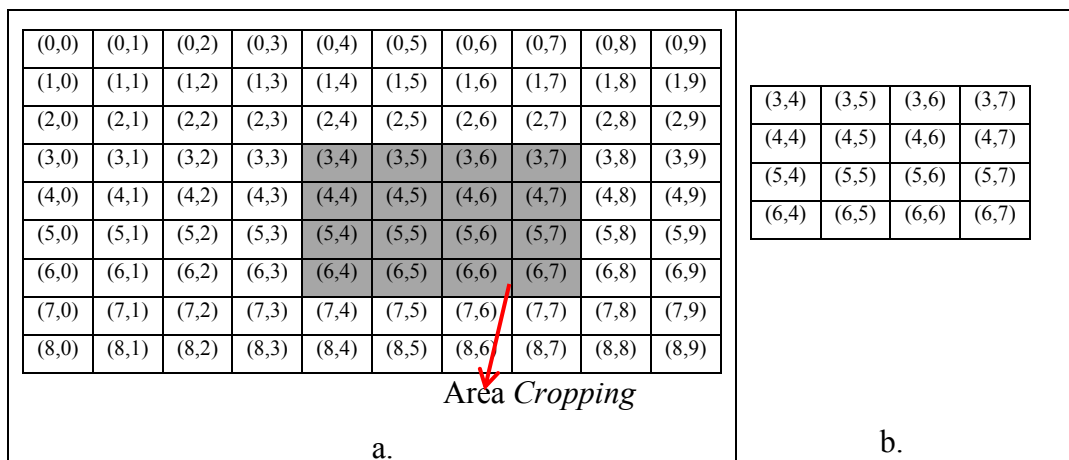
Tahap awal dalam proses klasifikasi citra multi daun adalah *preprocessing*. Pada tahap ini dilakukan *cropping* citra, *resize* citra, pemisahan *background* dan *foreground* serta segmentasi deteksi tepi dengan menggunakan operator *Canny*.

#### 4.1.1 *Cropping* Citra

Seperti yang telah dijelaskan pada Sub bab 3.1.2.1, *cropping* citra dilakukan untuk menghilangkan dan memotong bagian yang tidak diinginkan. Bagian yang tidak diinginkan tersebut terdapat di berbagai tepi dan daerah tertentu, sehingga penentuan bagian *cropping* citra tidak dapat dilakukan secara otomatis atau menggunakan satu perintah. Berikut merupakan perintah dari *cropping* citra Gambar 4.1.

```
Crop = imcrop(I,[967 0 3500 3456]);
```

*Cropping* citra dapat dilakukan berdasarkan titik koordinat, jumlah piksel atau hasil *zooming* daerah tertentu. Berikut merupakan ilustrasi *cropping* berdasarkan titik koordinat tertentu pada area citra. Dua koordinat digunakan untuk memotong bagian dari citra, yaitu koordinat awal yang merupakan titik koordinat awal bagi citra hasil *cropping* dan koordinat akhir yang merupakan titik koordinat akhir dari citra hasil *cropping*. Sehingga akan memberikan bentuk segiempat dengan setiap piksel yang ada pada area koordinat tertentu nantinya akan disimpan dalam citra yang baru.



Gambar 4.2 Proses *Cropping* Citra Berdasarkan Titik Koordinat. a. Citra Asli, b. Citra Hasil *Cropping*.

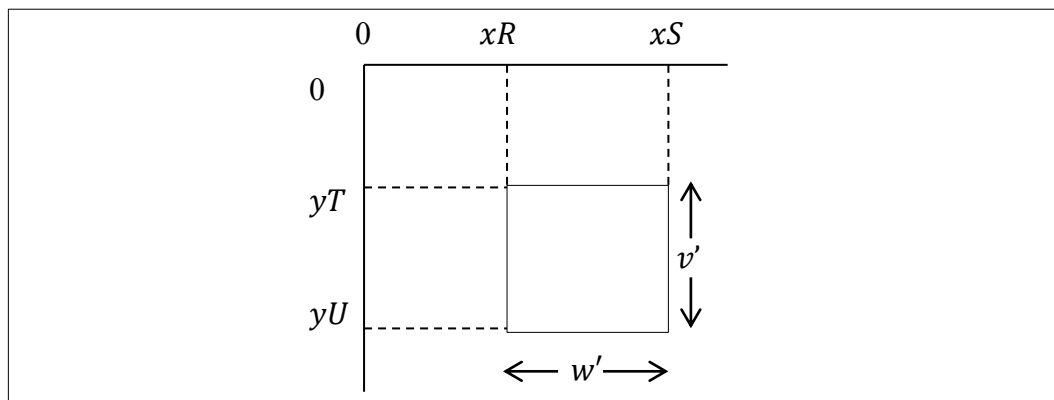
Berdasarkan Gambar 4.2 dijelaskan bahwa terjadi proses *cropping* citra dengan ukuran awal piksel citra asli yaitu  $10 \times 9$ , menjadi  $4 \times 4$  setelah dilakukan proses *cropping*. Pada proses *cropping* koordinat awal (3,4) dan koordinat akhir (6,7) atau dengan panjang 4 piksel dan lebar 4 piksel. Citra baru ini berisi nilai intensitas dari koordinat (3,4) sampai koordinat (6,7).

Ilustrasi *cropping* citra berdasarkan jumlah piksel disajikan pada Gambar 4.3.  $x$  merupakan panjang piksel citra asli dan  $y$  lebar piksel, sedangkan  $(xR, yT)$  dan  $(xS, yU)$  merupakan koordinat titik ujung kiri atas dan ujung kanan bawah citra yang dilakukan *cropping*, maka ukuran citra menjadi :

$$w' = xS - xR$$

$$v' = yU - yT$$

dimana  $w'$  merupakan panjang piksel daerah yang di-*cropping* dan  $v'$  merupakan lebar piksel daerah yang di-*cropping*.

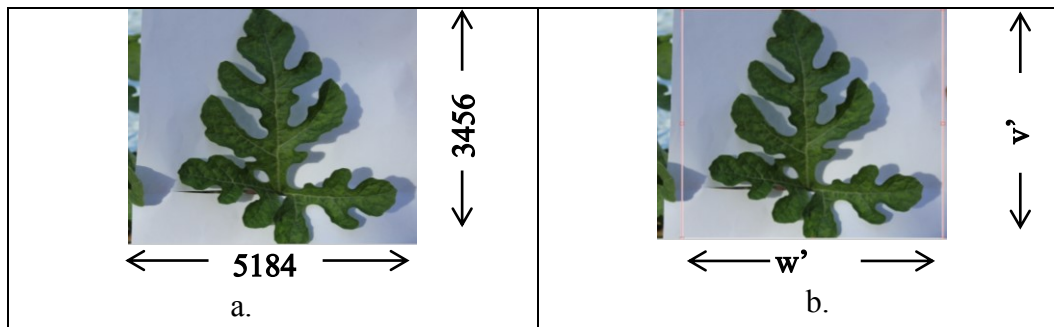


Gambar 4.3 *Cropping* Citra Berdasarkan Jumlah Piksel

Representasi proses *cropping* citra dari Gambar 4.1 tampak pada Gambar 4.4. Citra pada Gambar 4.1 berukuran  $5184 \times 3456$  sehingga dapat disimpulkan bahwa panjang piksel dari citra tersebut adalah 5185 ( $x = 5185$ ) dan lebar piksel dari citra tersebut adalah 3456 ( $y = 3456$ ). Bagian yang diinginkan terletak pada koordinat (967, 0) sampai (4467, 3456), dengan kata lain nilai  $xR = 967$ ,  $xS = 4467$ ,  $yT = 0$ , dan  $yU = 3456$ , sehingga diperoleh:

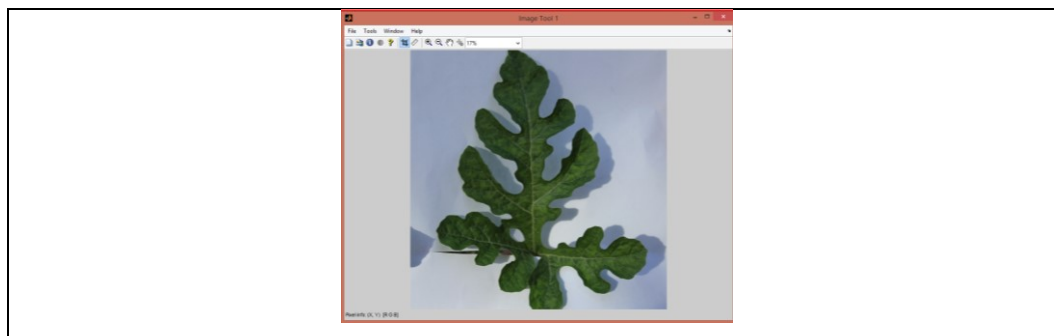
$$w' = xS - xR = 4467 - 967 = 3500$$

$$v' = yU - yT = 3456 - 0 = 3456$$



Gambar 4.4 Proses *Cropping* Citra Berdasarkan Jumlah Piksel. a. Citra Asli, b. Bagian Citra yang Ingin Dipotong

Dari proses tersebut diperoleh panjang citra baru ( $w'$ ) sebesar 3456 piksel dan lebar ( $v'$ ) 3500 piksel, sehingga dapat diperoleh citra baru dari hasil *cropping* dengan ukuran  $3456 \times 3500$  piksel.



Gambar 4.5 Hasil *Cropping* Citra

Namun perintah tersebut hanya bisa diberlakukan pada satu gambar saja, sehingga penentuan bagian *cropping* citra perlu dilakukan dengan menggunakan *tool* pada matlab dengan perintah sebagai berikut:

```
I = imread('IMG_0164.JPG');
imshow(I)
```

#### 4.1.2 *Resize* Citra

*Resize* citra adalah proses mengubah ukuran piksel citra dari ukuran besar menjadi ukuran yang lebih kecil, agar lebih mudah untuk dilakukan proses pendeteksian. Pada saat proses *cropping*, *size* yang dimiliki citra berbeda-beda dan masih berukuran besar sehingga dibutuhkan proses *resize* citra yang bertujuan



untuk membuat ukuran matriks piksel setiap data citra dengan ukuran yang sama. Berikut adalah ilustrasi proses *resize* baris dari citra *grayscale* yang berukuran  $10 \times 10$  yang akan di-*resize* menjadi berukuran  $5 \times 10$  dengan cara mencari nilai rata-rata dan nilai terdekat:

$$R = \begin{pmatrix} 160 & 161 & 164 & 81 & 74 & 156 & 166 & 165 & 168 & 168 \\ 160 & 163 & 155 & 59 & 50 & 121 & 166 & 171 & 173 & 175 \\ 163 & 166 & 102 & 76 & 61 & 62 & 135 & 174 & 186 & 178 \\ 166 & 156 & 88 & 51 & 56 & 95 & 77 & 135 & 195 & 177 \\ 168 & 116 & 54 & 95 & 61 & 76 & 56 & 150 & 200 & 185 \\ 167 & 148 & 66 & 64 & 62 & 74 & 137 & 194 & 200 & 192 \\ 152 & 83 & 67 & 56 & 58 & 135 & 187 & 182 & 197 & 195 \\ 152 & 83 & 50 & 56 & 64 & 127 & 80 & 72 & 89 & 158 \\ 189 & 191 & 161 & 83 & 40 & 96 & 94 & 69 & 97 & 161 \\ 199 & 198 & 200 & 169 & 90 & 99 & 184 & 154 & 173 & 159 \end{pmatrix}$$

Mencari rata-rata matriks R dari kolom pertama

$$\begin{array}{ccccc} 160 & 160 & 163 & 166 & 168 & 167 & 152 & 152 & 189 & 199 \\ & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow \\ & 160 & & 164 & & 167 & & 152 & & 194 \end{array}$$

Jadi, kolom pertama pada matriks R menjadi 160, 166, 167, 152, 199.

Mencari rata-rata matriks R dari kolom kedua

$$\begin{array}{ccccc} 161 & 163 & 166 & 156 & 116 & 148 & 83 & 83 & 191 & 198 \\ & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow \\ & 162 & & 161 & & 132 & & 83 & & 194 \end{array}$$

Jadi, kolom kedua pada matriks R menjadi 163, 166, 148, 83, 191.

Mencari rata-rata matriks R dari kolom ketiga

$$\begin{array}{ccccc} 164 & 155 & 102 & 88 & 54 & 66 & 67 & 50 & 161 & 200 \\ & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow \\ & 159 & & 95 & & 60 & & 58 & & 180 \end{array}$$

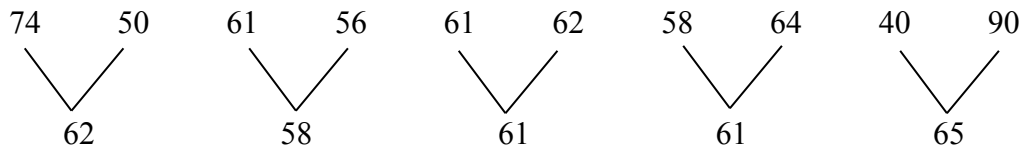
Jadi, kolom ketiga pada matriks R menjadi 155, 102, 66, 50, 161.

Mencari rata-rata matriks R dari kolom keempat

$$\begin{array}{ccccc} 81 & 59 & 76 & 51 & 95 & 64 & 56 & 56 & 83 & 169 \\ & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow & & \swarrow \searrow \\ & 70 & & 63 & & 79 & & 56 & & 126 \end{array}$$

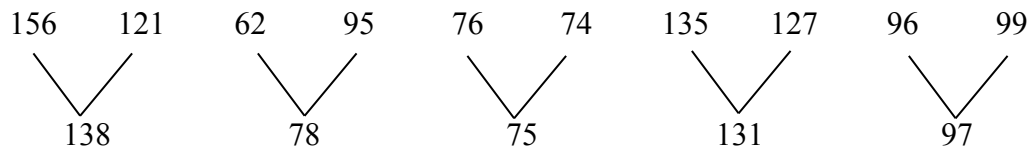
Jadi, kolom keempat pada matriks R menjadi 81, 51, 64, 56, 169.

Mencari rata-rata matriks R dari kolom kelima



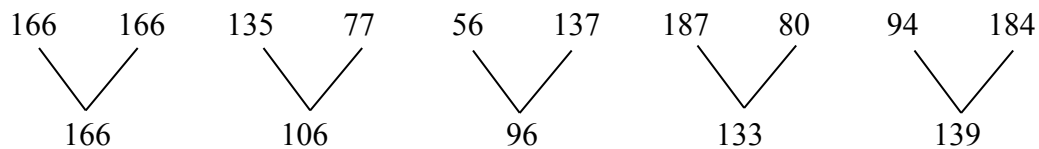
Jadi, kolom kelima pada matriks R menjadi 74, 56, 61, 64, 90.

Mencari rata-rata matriks R dari kolom keenam



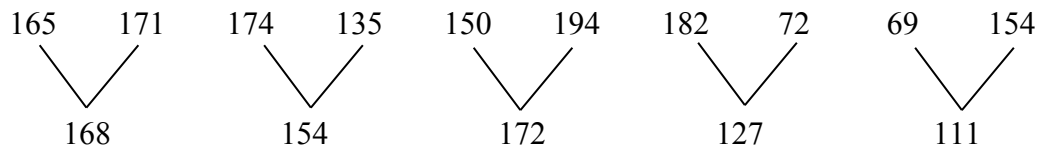
Jadi, kolom keenam pada matriks R menjadi 121, 62, 76, 135, 96.

Mencari rata-rata matriks R dari kolom ketujuh



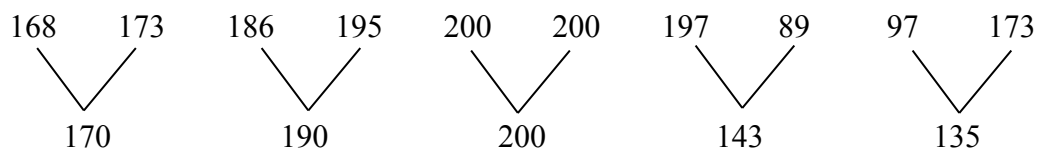
Jadi, kolom ketujuh pada matriks R menjadi 166, 135, 56, 80, 184.

Mencari rata-rata matriks R dari kolom kedelapan



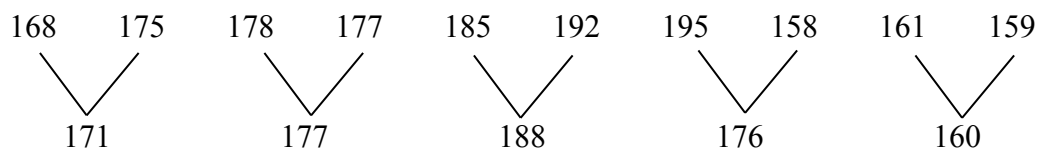
Jadi, kolom kedelapan pada matriks R menjadi 171, 135, 194, 182, 69.

Mencari rata-rata matriks R dari kolom kesembilan



Jadi, kolom kesembilan pada matriks R menjadi 168, 186, 200, 197, 173.

Mencari rata-rata matriks R dari kolom kesepuluh



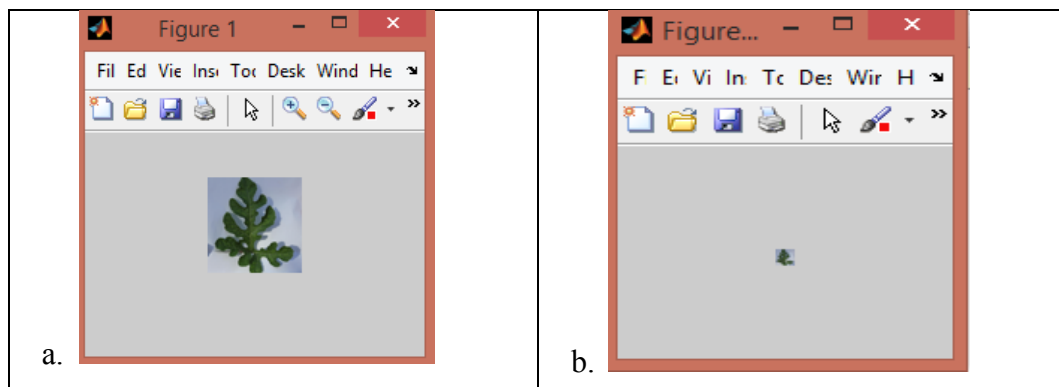
Jadi, kolom kesepuluh pada matriks R menjadi 168, 177, 185, 195, 161.

Sehingga matriks  $R$  yang diperoleh setelah proses *resize* menjadi matriks  $5 \times 10$  :

$$R = \begin{pmatrix} 160 & 163 & 155 & 81 & 74 & 121 & 166 & 171 & 168 & 168 \\ 166 & 166 & 102 & 51 & 56 & 62 & 135 & 135 & 186 & 177 \\ 167 & 148 & 66 & 64 & 61 & 76 & 56 & 194 & 200 & 185 \\ 152 & 83 & 50 & 56 & 64 & 135 & 135 & 182 & 197 & 195 \\ 199 & 191 & 161 & 169 & 90 & 96 & 184 & 69 & 173 & 161 \end{pmatrix}$$

Proses *resize* kolom dapat dilakukan melalui cara yang sama dengan *resize* baris, tetapi yang dicari adalah rata-rata antara dua piksel baris yang bertetangga dan mencari nilai terdekat.

Representasi *resize* citra dari Gambar 4.1 dengan *size*  $64 \times 64$  piksel dan  $10 \times 10$  piksel tampak pada Gambar 4.6.a dan 4.6.b.



Gambar 4.6 *Resize* Citra. a. Citra Berukuran  $64 \times 64$ , b. Citra Berukuran  $10 \times 10$

Data citra untuk pelatihan sebanyak 189 data citra yang terdiri dari 156 data citra daun semangka dan 33 data citra daun gulma. Selanjutnya dilakukan proses *resize* dengan ukuran 500 NaN. Berikut merupakan *listing* program untuk *resize* dari seluruh data pelatihan.

```
files = dir('*.jpg');
for n = 1 : 189

    filename = files(n).name;
    file = imread(filename);
    b = imresize(file, [500 NaN]);
    imwrite(b, ['resize', filename], 'jpg');

end
```

### 4.1.3 Pemisahan *Background* dan *Foreground* serta Menghilangkan Objek yang Tidak Diinginkan

Setelah dilakukan tahapan *cropping* dan *resize* pada citra terdapat bayangan di atas kertas putih sehingga dibutuhkan tahap pemisahan *background* dan *foreground*. Tujuannya adalah menjadikan *foreground* tetap berwarna RGB dan *background* berwarna putih (Eldahshan dkk (2014); Kulkarni dan Bhosane (2014)). Oleh karena objek yang digunakan dalam penelitian ini adalah daun maka objek selain daun dihilangkan. Proses ini disebut dengan memisahkan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan. Citra digital yang diambil adalah citra *Red Green Blue* (RGB). Tahap awal memisahkan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan adalah mengubah citra RGB menjadi citra *Hue Saturation Value* (HSV) lalu mengambil komponen warna H dan S dari citra HSV kemudian diubah menjadi citra hitam putih. Selanjutnya menegaskan citra hitam putih tersebut dan menggabungkan kedua citra tersebut sehingga menghasilkan sebuah citra dengan *background* berwarna hitam dan *foreground* berwarna putih serta objek yang tidak diinginkan hilang.

Ilustrasi dari pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan adalah sebagai berikut:

#### 1) Mengubah citra RGB menjadi citra HSV

Gunakan matriks piksel citra RGB dari Gambar 4.5.b. citra yang digunakan berukuran  $10 \times 10 \times 3$ .

(143,155,182)	(144,156,183)	(147,159,186)	(73,90,80)	(67,85,72)	(140,152,176)	(152,161,187)	(152,161,184)	(156,164,184)	(158,164,183)
(144,156,182)	(146,158,185)	(140,152,174)	(54,74,50)	(44,67,41)	(106,121,137)	(153,163,184)	(159,167,188)	(163,169,189)	(167,171,189)
(146,159,185)	(150,162,188)	(94,110,104)	(68,87,75)	(55,75,53)	(54,74,59)	(119,135,152)	(171,182,201)	(174,184,202)	(170,174,192)
(149,162,189)	(141,154,174)	(80,99,85)	(47,68,40)	(54,75,40)	(81,101,105)	(69,89,75)	(153,168,184)	(183,193,209)	(169,173,191)
(152,164,189)	(106,121,121)	(49,69,44)	(84,102,100)	(58,77,48)	(68,88,73)	(52,77,40)	(135,149,168)	(190,198,212)	(176,182,198)
(152,164,185)	(137,149,159)	(63,81,54)	(58,77,57)	(62,83,43)	(70,94,60)	(128,141,144)	(186,192,206)	(192,198,210)	(184,191,202)
(140,153,165)	(77,95,79)	(63,81,59)	(53,70,46)	(58,79,37)	(128,142,135)	(182,188,192)	(173,182,191)	(188,195,208)	(188,193,205)
(126,136,150)	(80,96,73)	(48,67,36)	(54,68,47)	(63,79,52)	(124,139,119)	(78,99,63)	(66,88,62)	(83,102,83)	(145,154,175)
(182,187,199)	(187,190,197)	(158,166,161)	(80,98,71)	(37,60,25)	(91,112,86)	(90,114,78)	(64,88,57)	(87,108,96)	(143,156,185)
(193,197,208)	(192,197,207)	(196,199,207)	(165,175,167)	(87,109,74)	(91,114,92)	(174,186,193)	(144,157,163)	(157,168,195)	(141,154,184)

dengan menggunakan rumus:

$$V = \frac{\max(R,G,B)}{255}$$

$$S = \begin{cases} 0, & \text{jika } V = 0 \\ \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)}, & \text{jika } V \neq 0 \end{cases}$$



$$H' = \begin{cases} \frac{60(G-B)}{\max(R,G,B)-\min(R,G,B)}, & \text{jika } \max(R,G,B) = R \text{ dan } G \geq B \\ 300 + \frac{60(B-G)}{\max(R,G,B)-\min(R,G,B)}, & \text{jika } \max(R,G,B) = R \text{ dan } B \geq G \\ 120 + \frac{60(B-R)}{\max(R,G,B)-\min(R,G,B)}, & \text{jika } \max(R,G,B) = G \\ 240 + \frac{60(R-G)}{\max(R,G,B)-\min(R,G,B)}, & \text{jika } \max(R,G,B) = B \end{cases}$$

$$H = \frac{H'}{360}$$

Untuk elemen pada baris pertama kolom pertama matriks RGB diperoleh matriks H, S, V sebagai berikut :

$$R = 143, G = 155, B = 182$$

$$V = \frac{\max(R,G,B)}{255} = \frac{182}{255} = 0.7137$$

$$S = \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)} = \frac{182-143}{182} = 0.2143$$

$$H' = 240 + \frac{60(R-G)}{\max(R,G,B)-\min(R,G,B)} = 240 + \frac{-720}{39} = 221.5384$$

$$H = \frac{H'}{360} = \frac{221.5384}{360} = 0.6154$$

Untuk elemen pada baris pertama kolom kedua matriks RGB diperoleh matriks H, S, V sebagai berikut:

$$R = 144, G = 156, B = 183$$

$$V = \frac{\max(R,G,B)}{255} = \frac{183}{255} = 0.7176$$

$$S = \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)} = \frac{183-144}{183} = 0.2131$$

$$H' = 240 + \frac{60(R-G)}{\max(R,G,B)-\min(R,G,B)} = 240 + \frac{-720}{39} = 221.5384$$

$$H = \frac{H'}{360} = \frac{221.5384}{360} = 0.6154$$

Untuk elemen pada baris kedua kolom pertama matriks RGB diperoleh matriks H, S, V sebagai berikut:

$$R = 144, G = 156, B = 182$$

$$V = \frac{\max(R,G,B)}{255} = \frac{182}{255} = 0.7137$$

$$S = \frac{\text{maks}(R,G,B) - \min(R,G,B)}{\text{maks}(R,G,B)} = \frac{182-144}{182} = 0.2088$$

$$H' = 240 + \frac{60(R-G)}{\text{maks}(R,G,B) - \min(R,G,B)} = 240 + \frac{-720}{38} = 221.0526$$

$$H = \frac{H'}{360} = \frac{221.0526}{360} = 0.6140$$

Untuk elemen pada baris kedua kolom kedua matriks RGB diperoleh matriks H, S, V sebagai berikut:

$$R = 146, G = 158, B = 185$$

$$V = \frac{\text{maks}(R,G,B)}{255} = \frac{185}{255} = 0.7255$$

$$S = \frac{\text{maks}(R,G,B) - \min(R,G,B)}{\text{maks}(R,G,B)} = \frac{185-146}{185} = 0.2108$$

$$H' = 240 + \frac{60(R-G)}{\text{maks}(R,G,B) - \min(R,G,B)} = 240 + \frac{-720}{39} = 221.5384$$

$$H = \frac{H'}{360} = \frac{221.0526}{360} = 0.6154$$

Untuk memperoleh elemen yang lain dari masing-masing matriks H, S, V dihitung dengan menggunakan rumus yang sama, sehingga seluruh komponen HSV dapat dilihat pada matriks komponen H, S, dan V berikut ini :

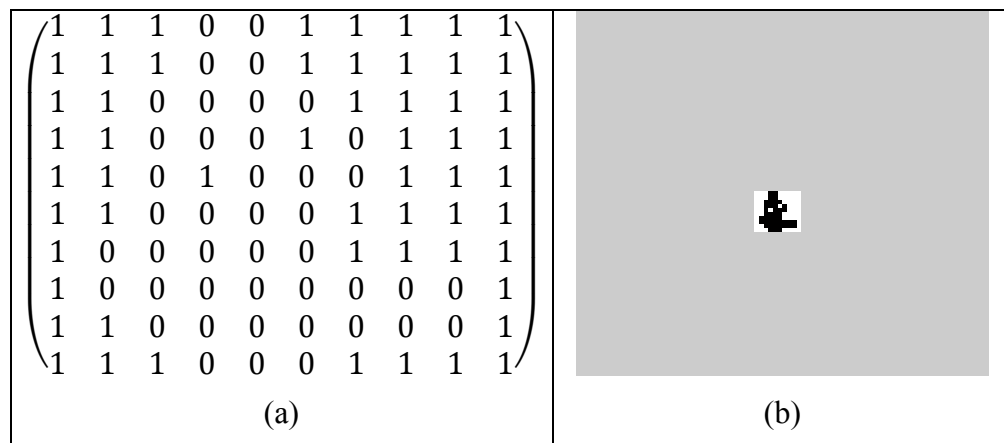
$$H = \begin{pmatrix} 0.6154 & 0.6154 & 0.6154 & 0.4020 & 0.3796 & 0.6111 & 0.6238 & 0.6198 & 0.6190 & 0.6267 \\ 0.6140 & 0.6154 & 0.6078 & 0.3056 & 0.3141 & 0.5860 & 0.6129 & 0.6207 & 0.6282 & 0.6364 \\ 0.6111 & 0.6140 & 0.4375 & 0.3947 & 0.3182 & 0.3750 & 0.5859 & 0.6056 & 0.6071 & 0.6364 \\ 0.6125 & 0.6010 & 0.3772 & 0.2917 & 0.2667 & 0.5278 & 0.3833 & 0.5860 & 0.6026 & 0.6364 \\ 0.6126 & 0.5000 & 0.3000 & 0.4815 & 0.2759 & 0.3750 & 0.2793 & 0.5960 & 0.6061 & 0.6212 \\ 0.6061 & 0.5758 & 0.2778 & 0.3250 & 0.2542 & 0.2843 & 0.5312 & 0.6167 & 0.6111 & 0.6019 \\ 0.5800 & 0.3519 & 0.3030 & 0.2847 & 0.2500 & 0.4167 & 0.5667 & 0.5833 & 0.6083 & 0.6176 \\ 0.5972 & 0.2826 & 0.2688 & 0.2778 & 0.2654 & 0.2917 & 0.2639 & 0.3077 & 0.3333 & 0.6167 \\ 0.6176 & 0.6167 & 0.3958 & 0.2778 & 0.2762 & 0.3013 & 0.2778 & 0.2957 & 0.4048 & 0.6151 \\ 0.6222 & 0.6111 & 0.6212 & 0.3667 & 0.2714 & 0.3406 & 0.5614 & 0.5526 & 0.6184 & 0.6163 \end{pmatrix}$$

$$S = \begin{pmatrix} 0.2143 & 0.2131 & 0.2097 & 0.1889 & 0.2118 & 0.2045 & 0.1872 & 0.1739 & 0.1522 & 0.1366 \\ 0.2088 & 0.2108 & 0.1954 & 0.3243 & 0.3881 & 0.2263 & 0.1685 & 0.1543 & 0.1376 & 0.1164 \\ 0.2108 & 0.2021 & 0.1455 & 0.2184 & 0.2933 & 0.2703 & 0.2171 & 0.1493 & 0.1386 & 0.1146 \\ 0.2116 & 0.1897 & 0.1919 & 0.4118 & 0.4667 & 0.2286 & 0.2247 & 0.1685 & 0.1244 & 0.1152 \\ 0.1958 & 0.1240 & 0.3623 & 0.1765 & 0.3766 & 0.2273 & 0.4805 & 0.1964 & 0.1038 & 0.1111 \\ 0.1784 & 0.1384 & 0.3333 & 0.2597 & 0.4819 & 0.3617 & 0.1111 & 0.0971 & 0.0857 & 0.0891 \\ 0.1515 & 0.1895 & 0.2716 & 0.3429 & 0.5316 & 0.0986 & 0.0521 & 0.0942 & 0.0962 & 0.0829 \\ 0.1600 & 0.2396 & 0.4627 & 0.3088 & 0.3418 & 0.1439 & 0.3636 & 0.2955 & 0.1863 & 0.1714 \\ 0.0854 & 0.0508 & 0.0482 & 0.2755 & 0.5833 & 0.2321 & 0.3158 & 0.3523 & 0.1944 & 0.2270 \\ 0.0721 & 0.0725 & 0.0531 & 0.0571 & 0.3211 & 0.2018 & 0.0984 & 0.1166 & 0.1949 & 0.2337 \end{pmatrix}$$

$$V = \begin{pmatrix} 0.7137 & 0.7176 & 0.7294 & 0.3529 & 0.3333 & 0.6902 & 0.7333 & 0.7216 & 0.7216 & 0.7176 \\ 0.7137 & 0.7255 & 0.6824 & 0.2902 & 0.2627 & 0.5373 & 0.7216 & 0.7373 & 0.7412 & 0.7412 \\ 0.7255 & 0.7373 & 0.4314 & 0.3412 & 0.2941 & 0.2902 & 0.5961 & 0.7882 & 0.7922 & 0.7529 \\ 0.7412 & 0.6824 & 0.3882 & 0.2667 & 0.2941 & 0.4118 & 0.3490 & 0.7216 & 0.8196 & 0.7490 \\ 0.7412 & 0.4745 & 0.2706 & 0.4000 & 0.3020 & 0.3451 & 0.3020 & 0.6588 & 0.8314 & 0.7765 \\ 0.7255 & 0.6235 & 0.3176 & 0.3020 & 0.3255 & 0.3686 & 0.5647 & 0.8078 & 0.8235 & 0.7922 \\ 0.6471 & 0.3725 & 0.3176 & 0.2745 & 0.3098 & 0.5569 & 0.7529 & 0.7490 & 0.8157 & 0.8039 \\ 0.5882 & 0.3765 & 0.2627 & 0.2667 & 0.3098 & 0.5451 & 0.3882 & 0.3451 & 0.4000 & 0.6863 \\ 0.7804 & 0.7725 & 0.6510 & 0.3843 & 0.2353 & 0.4392 & 0.4471 & 0.3451 & 0.4235 & 0.7255 \\ 0.8157 & 0.8118 & 0.8118 & 0.6863 & 0.4275 & 0.4471 & 0.7569 & 0.6392 & 0.7647 & 0.7216 \end{pmatrix}$$

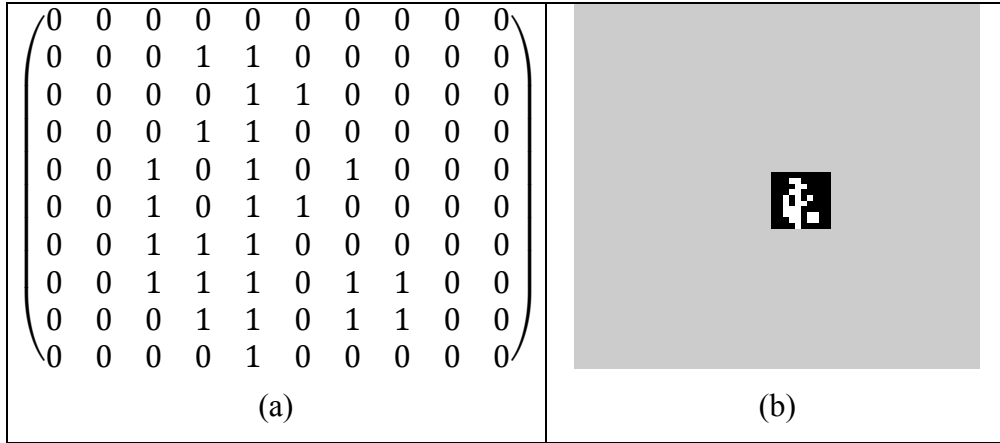
- 2) Mengambil komponen warna H dan S dari HSV kemudian mengubah komponen tersebut menjadi hitam putih.

Nilai elemen matriks komponen *Hue* yang bernilai lebih besar atau sama dengan 0.5 adalah 1. Sedangkan yang bernilai kurang dari 0.5 bernilai 0. Sehingga matriks yang diperoleh dari mengubah komponen *Hue* menjadi citra hitam putih dan citra yang dihasilkan tampak pada Gambar 4.7.



Gambar 4.7 (a) Matriks Hitam Putih Komponen *Hue*, (b) Citra Hitam Putih Komponen *Hue*.

Nilai elemen matriks komponen *Saturation* yang bernilai lebih besar atau sama dengan 0.5 adalah 1. Sedangkan yang bernilai kurang dari 0.5 bernilai 0. Hasil matriks yang diperoleh dari mengubah komponen *Saturation* menjadi citra hitam putih dan citra yang dihasilkan tampak pada Gambar 4.8.



Gambar 4.8 (a) Matriks Hitam Putih Komponen *Saturation*, (b) Citra Hitam Putih Komponen *Saturation*.

3) Menegasikan komponen citra.

Citra yang diperoleh dari proses membuat citra hitam putih pada Gambar 4.7.(a) (komponen *Hue*) adalah citra dengan *background* berwarna putih dan *foreground* berwarna hitam, sehingga perlu untuk menegasikan citra yaitu mengubah nilai 0 menjadi 1 dan begitu pula sebaliknya. Matriks  $H_N$  merupakan matriks hitam putih komponen *Hue* yang telah dinegasikan.

$$H_N = \begin{pmatrix}
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0
 \end{pmatrix}$$

Hal yang sama berlaku untuk citra hitam putih pada komponen *Saturation* sehingga matriks komponen *Saturation* menjadi:

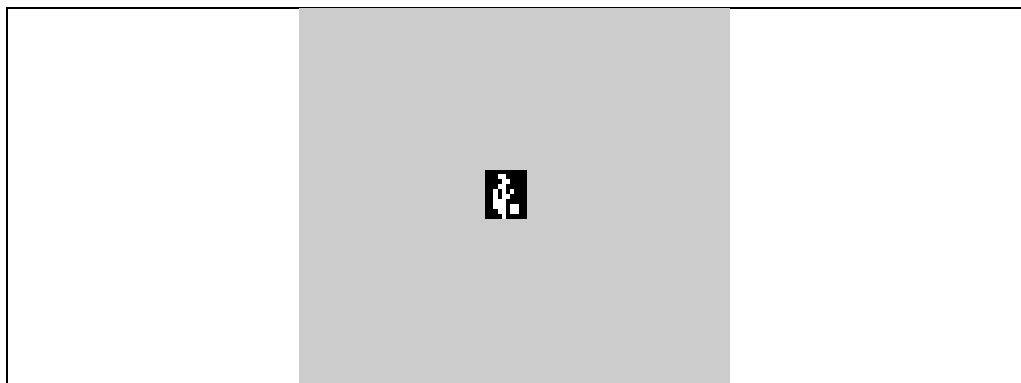
$$S_N = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Matriks  $S_N$  merupakan matriks hitam putih komponen *Saturation* yang telah dinegasikan.

4) Menggabungkan komponen *Hue* dan *Saturation*.

Cara menggabungkan matriks komponen *Hue* dan *Saturation* yaitu menggunakan logika AND. Matriks penggabungan komponen *Hue* dan *Saturation* adalah sebagai berikut:

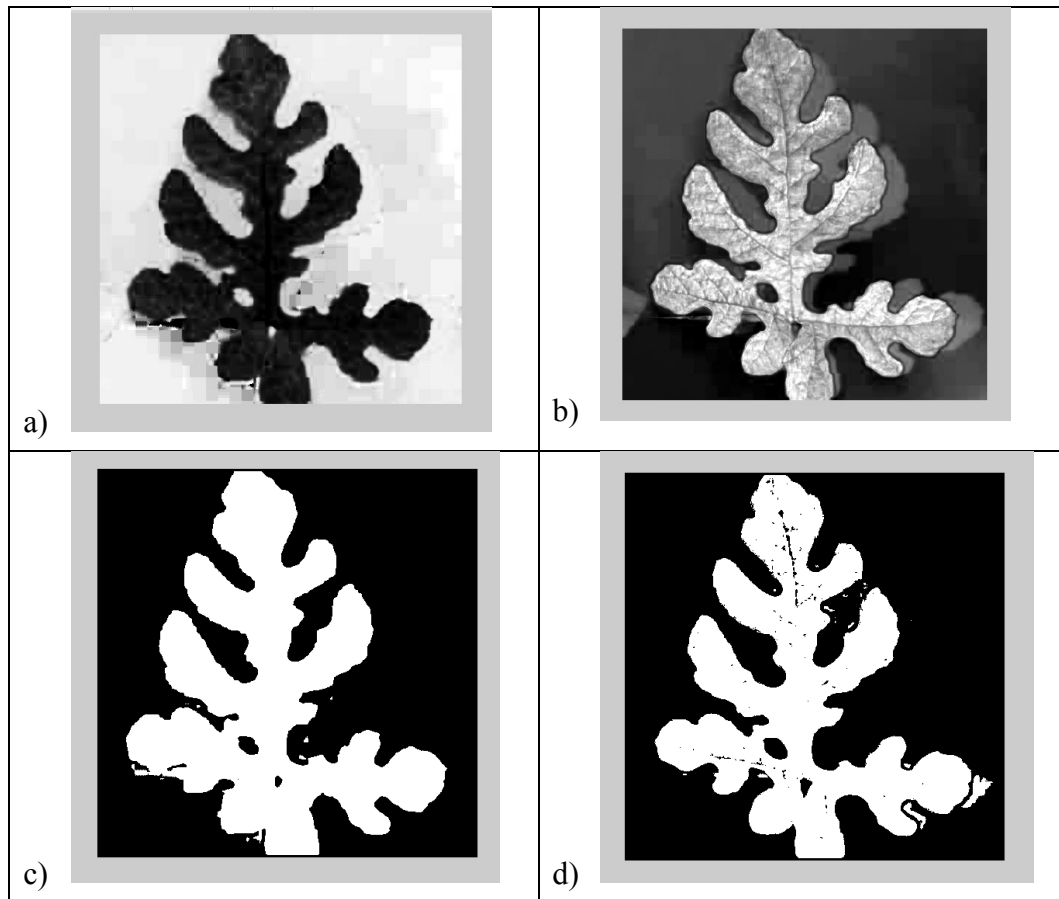
$$HS = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Gambar 4.9 Hasil Pemisahan *Background* dan *Foreground* untuk Citra Berukuran  $10 \times 10$ .

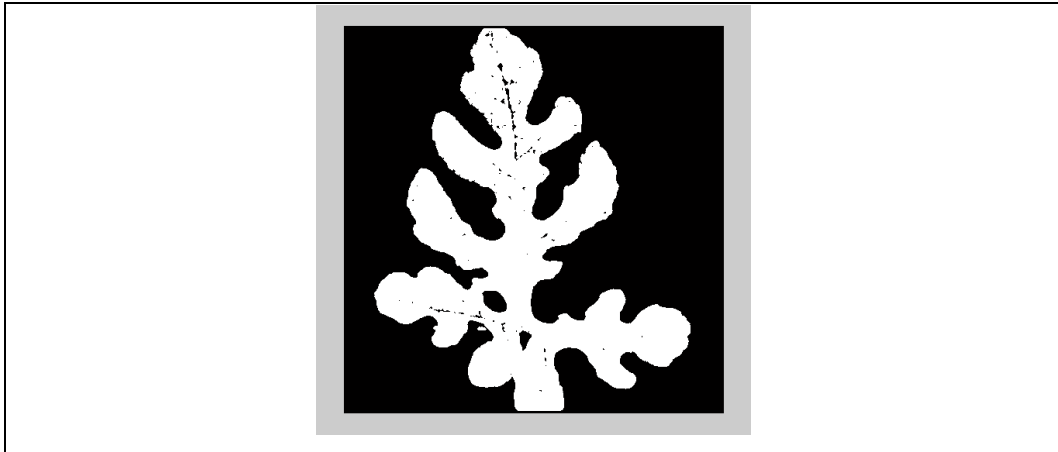


Berikut representasi proses memisahkan *background* dengan citra yang berukuran 500 NaN.



Gambar 4.10 Proses Pemisahan *Background* dan *Foreground* serta Menghilangkan Objek yang Tidak Diinginkan. a) Pengambilan Komponen Warna *Hue*, b) Pengambilan Komponen Warna *Saturation*, c) Pengubahan Komponen Warna *Hue* menjadi Hitam Putih yang Telah Dinegasikan, dan d) Pengubahan Komponen Warna *Saturation* menjadi Citra Hitam Putih yang Telah Dinegasikan.

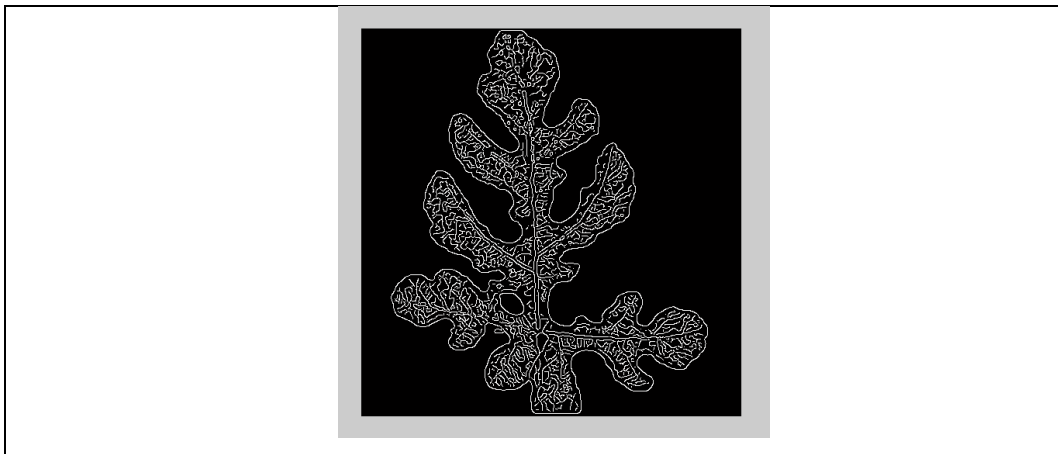
Setelah komponen warna *Hue* dan *Saturation* menjadi citra hitam putih selanjutnya kedua citra tersebut digabung menjadi sebuah citra dan noise yang ada dihilangkan sehingga hasil yang diperoleh tampak seperti pada Gambar 4.11. Citra ini digunakan untuk mencari nilai ekstraksi fitur bentuk.



Gambar 4.11. Penggabungan Citra Hitam Putih dari Komponen Warna *Hue* dan *Saturation*

#### 4.1.4 Segmentasi Citra dengan Menggunakan Deteksi Tepi

Segmentasi citra dengan menggunakan deteksi tepi diperlukan untuk mencari informasi nilai fitur tekstur. Operator yang digunakan dalam deteksi tepi ini adalah operator *canny* yang didapatkan dari citra hasil kali antara citra asli yang di konversi menjadi citra *grayscale* dengan citra hasil pemisahan *background*. Citra hasil deteksi tepi dengan operator *canny* ditampilkan pada Gambar 4.12.



Gambar 4.12 Segmentasi dengan Metode Deteksi Tepi Operator *Canny*

Representasi proses deteksi tepi dengan operator *canny* menggunakan Gambar 4.6.b yang merupakan citra RGB hasil *resize* menjadi berukuran  $10 \times 10$

piksel, citra tersebut kemudian diubah dari citra RGB menjadi citra *grayscale* dengan persamaan :

$$Gray = 0.299 * R + 0.587 * G + 0.114 * B$$

Berdasarkan persamaan tersebut maka diperoleh citra *grayscale* sebagai berikut :

Untuk elemen pada baris pertama kolom pertama matriks HS diperoleh matriks *Grayscale* sebagai berikut :

$$\begin{aligned} Gray &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ &= (0.229 \times 143) + (0.587 \times 155) + (0.114 \times 182) = 154.49 = 154 \end{aligned}$$

Untuk elemen pada baris pertama kolom kedua matriks HS diperoleh matriks *Grayscale* sebagai berikut :

$$\begin{aligned} Gray &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ &= (0.229 \times 144) + (0.587 \times 156) + (0.114 \times 183) = 155.49 = 155 \end{aligned}$$

Untuk elemen pada baris pertama kolom ketiga matriks HS diperoleh matriks *Grayscale* sebagai berikut :

$$\begin{aligned} Gray &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ &= (0.229 \times 147) + (0.587 \times 159) + (0.114 \times 186) = 158.49 = 158 \end{aligned}$$

Untuk elemen pada baris kedua kolom pertama matriks HS diperoleh matriks *Grayscale* sebagai berikut :

$$\begin{aligned} Gray &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ &= (0.229 \times 144) + (0.587 \times 156) + (0.114 \times 182) = 155.376 = 155 \end{aligned}$$

Untuk elemen pada baris kedua kolom kedua matriks HS diperoleh matriks *Grayscale* sebagai berikut :

$$\begin{aligned} Gray &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ &= (0.229 \times 146) + (0.587 \times 158) + (0.114 \times 185) = 157.27 = 157 \end{aligned}$$

Untuk memperoleh elemen yang lain dari matriks *Gray* dihitung dengan menggunakan rumus yang sama. Elemen pada matriks *Gray* yang dihasilkan

awalnya berupa bilangan pecahan, selanjutnya dilakukan proses pembulatan untuk mendapatkan bilangan bulat. Pembulatan ini menggunakan fungsi *fix* yang prosesnya mengabaikan angka pecahan yang dimiliki meskipun angka tersebut mendekati nilai 1. Seluruh komponen *grayscale* dapat dilihat pada matriks *Gray* berikut ini :

$$Gray = \begin{pmatrix} 154 & 155 & 158 & 84 & 78 & 151 & 161 & 161 & 164 & 164 \\ 155 & 157 & 151 & 65 & 57 & 118 & 162 & 167 & 169 & 172 \\ 158 & 161 & 105 & 80 & 67 & 66 & 132 & 181 & 183 & 175 \\ 161 & 152 & 92 & 59 & 65 & 95 & 81 & 165 & 192 & 174 \\ 163 & 117 & 60 & 96 & 68 & 80 & 65 & 147 & 197 & 182 \\ 163 & 147 & 73 & 69 & 72 & 83 & 137 & 192 & 198 & 190 \\ 150 & 88 & 73 & 62 & 68 & 137 & 187 & 180 & 194 & 193 \\ 135 & 89 & 58 & 61 & 71 & 132 & 89 & 78 & 94 & 154 \\ 187 & 190 & 163 & 90 & 49 & 103 & 103 & 77 & 100 & 155 \\ 197 & 197 & 199 & 171 & 98 & 105 & 183 & 154 & 168 & 154 \end{pmatrix}$$

Citra yang digunakan untuk deteksi tepi adalah citra hasil kali antara citra *grayscale* dan citra hasil pemisahan *background* (*T*).

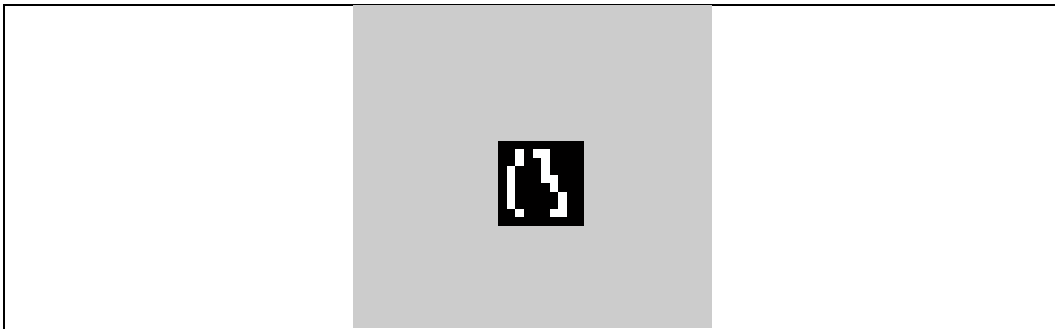
$$T = Gray \cdot HS$$

$$= \begin{pmatrix} 154 & 155 & 158 & 84 & 78 & 151 & 161 & 161 & 164 & 164 \\ 155 & 157 & 151 & 65 & 57 & 118 & 162 & 167 & 169 & 172 \\ 158 & 161 & 105 & 80 & 67 & 66 & 132 & 181 & 183 & 175 \\ 161 & 152 & 92 & 59 & 65 & 95 & 81 & 165 & 192 & 174 \\ 163 & 117 & 60 & 96 & 68 & 80 & 65 & 147 & 197 & 182 \\ 163 & 147 & 73 & 69 & 72 & 83 & 137 & 192 & 198 & 190 \\ 150 & 88 & 73 & 62 & 68 & 137 & 187 & 180 & 194 & 193 \\ 135 & 89 & 58 & 61 & 71 & 132 & 89 & 78 & 94 & 154 \\ 187 & 190 & 163 & 90 & 49 & 103 & 103 & 77 & 100 & 155 \\ 197 & 197 & 199 & 171 & 98 & 105 & 183 & 154 & 168 & 154 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 65 & 57 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 67 & 66 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 59 & 65 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60 & 0 & 68 & 0 & 65 & 0 & 0 & 0 \\ 0 & 0 & 73 & 0 & 72 & 83 & 0 & 0 & 0 & 0 \\ 0 & 0 & 73 & 62 & 68 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 58 & 61 & 71 & 0 & 89 & 78 & 0 & 0 \\ 0 & 0 & 0 & 90 & 49 & 0 & 103 & 77 & 0 & 0 \\ 0 & 0 & 0 & 0 & 98 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Matriks *T* kemudian dideteksi tepi dengan menggunakan operator *canny*. Gambar deteksi tepi diperlihatkan pada Gambar 4.13.

$$Canny = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Gambar 4.13 Deteksi Tepi Berukuran  $10 \times 10$  piksel

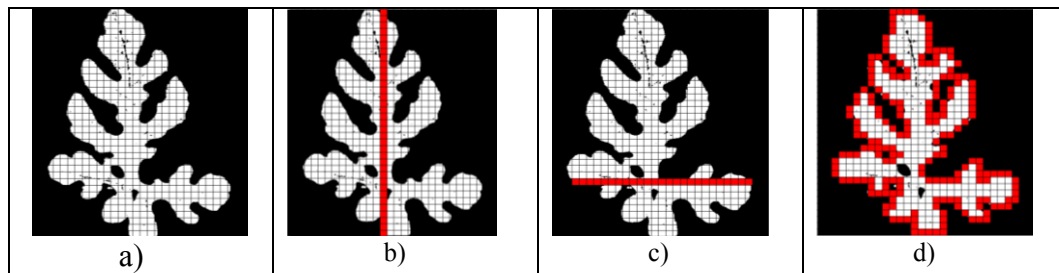
## 4.2 Ekstraksi Fitur

Fitur yang digunakan untuk mengklasifikasikan daun semangka dan gulma ada dua yaitu fitur bentuk dan tekstur.

### 4.2.1 Fitur Bentuk

Citra yang digunakan untuk mencari fitur bentuk adalah citra hasil pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan. Pada fitur bentuk informasi yang dicari terlebih dahulu adalah informasi mengenai diameter, panjang fisiologis, lebar fisiologis, luas daun, dan perimeter daun. Informasi ini dibutuhkan untuk mendapatkan informasi aspek rasio, faktor *form*, *rectangularity*, faktor *narrow*, rasio *perimeter* dari diameter, dan rasio dari panjang fisiologis dan lebar fisiologis.

Dalam perhitungan fitur bentuk citra yang digunakan adalah citra hitam putih hasil pemisahan *background* dan *foreground*. Piksel yang diperhatikan adalah piksel yang berwarna putih atau yang bernilai 1. Ilustrasi perhitungannya adalah sebagai berikut :



Gambar 4.14 Proses Ekstraksi Bentuk. a) Area, b) Panjang Fisiologis, c) Lebar Fisiologis, dan d) Perimeter.

- a) Area = total jumlah piksel selain nol yang merupakan objek dalam citra.
- b) Panjang fisiologis = jumlah panjang maksimum piksel (pada panjang daun).
- c) Lebar fisiologis = jumlah lebar maksimum semua piksel (pada lebar daun).
- d) Diameter = Skalar yang menentukan diameter lingkaran dengan area yang

sama dengan daerah. Dihitung dengan rumus  $\sqrt{\frac{4 \text{ Area}}{\pi}}$ .

- e) Perimeter = jumlah piksel di sekitar batas bentuk objek.

Informasi area, panjang fisiologis, lebar fisiologis, diameter, dan perimeter dapat digunakan untuk mencari informasi aspek rasio, faktor *form*, *rectangularity*, faktor *narrow*, rasio *perimeter* dari diameter, dan rasio dari panjang fisiologis dan lebar fisiologis dengan menggunakan persamaan 2.4 sampai persamaan 2.9.

Kode program untuk mendapatkan informasi tersebut adalah sebagai berikut:

```
diameter = stats(idx).EquivDiameter;
panjang_fisiologis = stats(idx).MajorAxisLength;
lebar_fisiologis = stats(idx).MinorAxisLength;
area = stats(idx).Area;
perimeter = stats(idx).Perimeter;

aspek_rasio = panjang_fisiologis/lebar_fisiologis;
faktor_form = (4*pi*area)/perimeter^2;
rectangularity = panjang_fisiologis*lebar_fisiologis/area;
faktor_narrow = diameter/panjang_fisiologis;
rasio_perimeter_dari_diameter = perimeter/diameter;
rasio_perimeter_dari_panjang_dan_lebar_fisiologi =
perimeter/(panjang_fisiologis+lebar_fisiologis);
```

Nilai fitur bentuk dengan menggunakan kode program tersebut untuk data citra tunggal disajikan pada Lampiran 1.

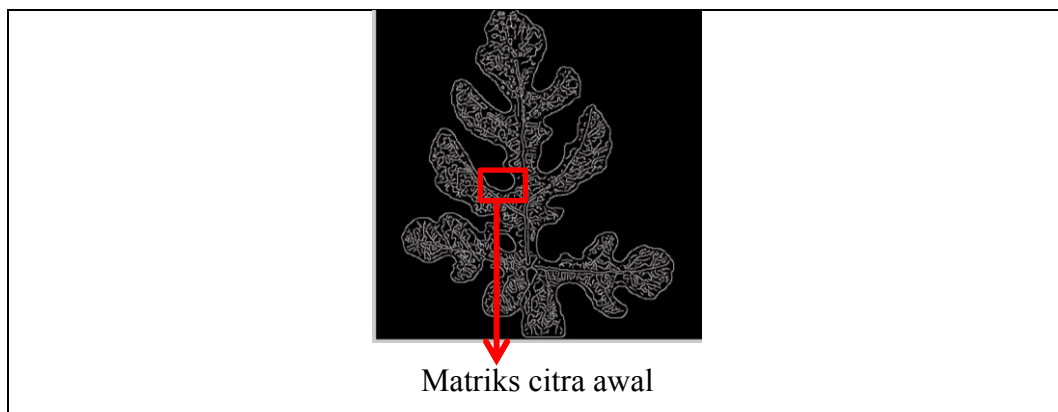


#### 4.2.2 Fitur Tekstur

Citra yang digunakan dalam metode *Gray Level Co-occurrence Matrices* (GLCM) dalam penelitian ini adalah citra biner yang merupakan gabungan citra hitam putih dari komponen warna *Hue* dan *Saturation* yang bernilai 0 dan 1.

Level yang digunakan pada penelitian ini adalah 2 dan menggunakan 4 sudut orientasi yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . Karena level yang digunakan adalah 2 maka matriks yang diperoleh adalah matriks  $2 \times 2$  dengan jarak 1 ketetanggaan.

Implementasi metode *Gray Level Co-Occurrence Matrices* (GLCM) menggunakan citra hasil deteksi tepi dengan ilustrasi sebagai berikut:



Gambar 4.15 Matriks Citra Awal untuk Perhitungan GLCM

$$\text{Matriks citra awal} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

##### 1) Filter Horizontal ( $0^\circ$ )

Tahapan dalam mencari filter horizontal GLCM adalah sebagai berikut:

a) Menentukan arah perhitungan GLCM dari matriks citra awal

$$\begin{pmatrix} 0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 1 \rightarrow 1 \\ 0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 0 \\ 0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 0 \\ 1 \rightarrow 1 \rightarrow 1 \rightarrow 0 \rightarrow 0 \rightarrow 0 \\ 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 0 \rightarrow 0 \\ 0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 0 \end{pmatrix}$$

b) Menentukan *Count Matrix*.

*Count matrix* merupakan matriks  $2 \times 2$  dengan 2 *gray tone* yang nilainya 0 dan 1. Matriks ini diperoleh dari menghitung jumlah *gray tone i* dan *j* yang memiliki ketetanggaan yang sesuai.

<i>Gray tone</i>	0	1
0	(0,0)	(0,1)
1	(1,0)	(1,1)

Sehingga *gray tone* dari matriks citra asli didapatkan sebagai berikut :

<i>Gray tone</i>	0	1
0	13	5
1	5	4

Sehingga didapatkan :

$$GLCM = \begin{pmatrix} 13 & 5 \\ 5 & 4 \end{pmatrix}$$

c) Membuat GLCM simetri:

i. *Transpose GLCM* =  $\begin{pmatrix} 13 & 5 \\ 5 & 4 \end{pmatrix}$

ii. Jumlahkan *GLCM* dan *transpose GLCM*

$$\begin{pmatrix} 13 & 5 \\ 5 & 4 \end{pmatrix} + \begin{pmatrix} 13 & 5 \\ 5 & 4 \end{pmatrix} = \begin{pmatrix} 26 & 10 \\ 10 & 8 \end{pmatrix}$$

d) Menghitung total keseluruhan nilai matriks.

Total keseluruhan nilai matriks adalah 54

e) Menentukan matriks Normalisasi

Membagi setiap elemen matriks dengan total nilai keseluruhan matriks.

$$\text{Normalisasi} = \begin{pmatrix} \frac{26}{54} & \frac{10}{54} \\ \frac{10}{54} & \frac{8}{54} \end{pmatrix} = \begin{pmatrix} 0.48 & 0.185 \\ 0.185 & 0.148 \end{pmatrix}$$

$$P_0(i,j) = \begin{pmatrix} 0.48 & 0.185 \\ 0.185 & 0.148 \end{pmatrix}$$

f) Setelah didapatkan matriks  $P_0(i,j)$ , selanjutnya mencari nilai *contrast*, *correlation*, *energy* dan *homogeneity*.

$$\begin{aligned}
Contrast &= \sum_{i,j} (i-j)^2 P_0(i,j) \\
&= (0-0)^2 0.48 + (0-1)^2 0.185 + (1-0)^2 0.185 \\
&\quad + (1-1)^2 0.148 \\
&= 0.37
\end{aligned}$$

$$\begin{aligned}
\mu_i &= \sum_{i,j} i P_0(i,j) \\
&= (0)0.48 + (1)0.185 + (0)0.185 + (1)0.148 \\
&= 0.333
\end{aligned}$$

$$\begin{aligned}
\mu_j &= \sum_{i,j} j P_0(i,j) \\
&= (0)0.48 + (1)0.185 + (0)0.185 + (1)0.148 \\
&= 0.333
\end{aligned}$$

$$\begin{aligned}
\sigma_i^2 &= \sum_{i,j} (i - \mu_i)^2 P_0(i,j) \\
&= (0 - 0.333)^2 0.48 + (1 - 0.333)^2 0.185 \\
&\quad + (0 - 0.333)^2 0.185 + (1 - 0.333)^2 0.148 \\
&= 0.2219
\end{aligned}$$

$$\begin{aligned}
\sigma_j^2 &= \sum_{i,j} (j - \mu_j)^2 P_0(i,j) \\
&= (0 - 0.333)^2 0.48 + (1 - 0.333)^2 0.185 \\
&\quad + (0 - 0.333)^2 0.185 + (1 - 0.333)^2 0.148 \\
&= 0.2219
\end{aligned}$$

$$\begin{aligned}
Correlation &= \sum_{i,j} P_0(i,j) \left[ \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \\
&= 0.48 \left[ \frac{(0-0.333)(0-0.333)}{\sqrt{(0.2219)(0.2219)}} \right] + 0.185 \left[ \frac{(1-0.333)(0-0.333)}{\sqrt{(0.2219)(0.2219)}} \right] \\
&\quad + 0.185 \left[ \frac{(0-0.333)(1-0.333)}{\sqrt{(0.2219)(0.2219)}} \right] + 0.148 \left[ \frac{(1-0.333)(1-0.333)}{\sqrt{(0.2219)(0.2219)}} \right] \\
&= 0.48(0.4997) + 0.185(-1.0009) + 0.185(-1.0009) \\
&\quad + 0.148(2.0049) \\
&= 0.1662
\end{aligned}$$

$$\begin{aligned}
 Energy &= \sum_{i,j} P_0(i,j)^2 \\
 &= 0.48^2 + 0.185^2 + 0.185^2 + 0.148^2 \\
 &= 0.3208
 \end{aligned}$$

$$\begin{aligned}
 Homogeneity &= \sum_{i,j} \frac{P_0(i,j)}{1+(i-j)^2} \\
 &= \frac{0.48}{1+(0-0)^2} + \frac{0.185}{1+(0-1)^2} + \frac{0.185}{1+(1-0)^2} + \frac{0.148}{1+(1-1)^2} \\
 &= 0.48 + 0.0925 + 0.0925 + 0.148 \\
 &= 0.813
 \end{aligned}$$

## 2) Filter Diagonal Kanan Atas Kiri Bawah (45°)

Tahapan dalam mencari filter horizontal GLCM adalah sebagai berikut:

a) Menentukan arah perhitungan GLCM dari matriks citra awal

$$\begin{pmatrix}
 0 & 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 \\
 1 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0
 \end{pmatrix}$$

b) Menentukan GLCM.

GLCM merupakan matriks  $2 \times 2$  dengan 2 *gray tone* yang nilainya 0 dan 1. Matriks ini diperoleh dari menghitung jumlah *gray tone*  $i$  dan  $j$  yang memiliki ketetanggaan yang sesuai.

<i>Gray tone</i>	0	1
0	(0,0)	(0,1)
1	(1,0)	(1,1)

Sehingga *gray tone* dari matriks citra asli didapatkan sebagai berikut :

<i>Gray tone</i>	0	1
0	12	5
1	6	2

Sehingga didapatkan :

$$GLCM = \begin{pmatrix} 12 & 5 \\ 6 & 2 \end{pmatrix}$$

c) Membuat GLCM simetri:

i. *Transpose* GLCM =  $\begin{pmatrix} 12 & 6 \\ 5 & 2 \end{pmatrix}$

ii. Jumlahkan GLCM dan *transpose* GLCM

$$\begin{pmatrix} 12 & 5 \\ 6 & 2 \end{pmatrix} + \begin{pmatrix} 12 & 6 \\ 5 & 2 \end{pmatrix} = \begin{pmatrix} 24 & 11 \\ 11 & 4 \end{pmatrix}$$

d) Menghitung total keseluruhan nilai matriks.

Total keseluruhan nilai matriks adalah 50

e) Menentukan matriks Normalisasi

Membagi setiap elemen matriks dengan total nilai keseluruhan matriks.

$$\text{Normalisasi} = \begin{pmatrix} \frac{24}{50} & \frac{11}{50} \\ \frac{11}{50} & \frac{4}{50} \end{pmatrix} = \begin{pmatrix} 0.48 & 0.22 \\ 0.22 & 0.08 \end{pmatrix}$$

$$P_{45}(i, j) = \begin{pmatrix} 0.48 & 0.22 \\ 0.22 & 0.08 \end{pmatrix}$$

f) Setelah didapatkan matriks  $P_{45}(i, j)$ , selanjutnya mencari nilai *contrast*, *correlation*, *energy* dan *homogeneity*.

$$\begin{aligned} \text{Contrast} &= \sum_{i,j} (i - j)^2 P_{45}(i, j) \\ &= (0 - 0)^2 0.48 + (0 - 1)^2 0.22 + (1 - 0)^2 0.22 \\ &\quad + (1 - 1)^2 0.08 = 0.44 \end{aligned}$$

$$\begin{aligned} \mu_i &= \sum_{i,j} i P_{45}(i, j) \\ &= (0)0.48 + (1)0.22 + (0)0.22 + (1)0.08 \\ &= 0.3 \end{aligned}$$

$$\begin{aligned} \mu_j &= \sum_{i,j} j P_{45}(i, j) \\ &= (0)0.48 + (1)0.22 + (0)0.22 + (1)0.08 \\ &= 0.3 \end{aligned}$$

$$\begin{aligned} \sigma_i^2 &= \sum_{i,j} (i - \mu_i)^2 P_{45}(i, j) \\ &= (0 - 0.3)^2 0.48 + (1 - 0.3)^2 0.22 + (0 - 0.3)^2 0.22 \\ &\quad + (1 - 0.3)^2 0.08 \\ &= 0.21 \end{aligned}$$

$$\begin{aligned}
\sigma_j^2 &= \sum_{i,j} (j - \mu_j)^2 P_{45}(i,j) \\
&= (0 - 0.3)^2 0.48 + (1 - 0.3)^2 0.22 + (0 - 0.3)^2 0.22 \\
&\quad + (1 - 0.3)^2 0.08 \\
&= 0.21
\end{aligned}$$

$$\begin{aligned}
Correlation &= \sum_{i,j} P_{45}(i,j) \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \\
&= 0.48 \left[ \frac{(0-0.3)(0-0.3)}{\sqrt{(0.21)(0.21)}} \right] + 0.22 \left[ \frac{(1-0.3)(0-0.3)}{\sqrt{(0.21)(0.21)}} \right] \\
&\quad + 0.2 \left[ \frac{(0-0.3)(1-0.3)}{\sqrt{(0.21)(0.21)}} \right] + 0.08 \left[ \frac{(1-0.3)(1-0.3)}{\sqrt{(0.21)(0.21)}} \right] \\
&= 0.48(0.4286) + 0.22(-1) + 0.22(-1) \\
&\quad + 0.08(2.3333) \\
&= 0.024
\end{aligned}$$

$$\begin{aligned}
Energy &= \sum_{i,j} P_{45}(i,j)^2 \\
&= 0.48^2 + 0.22^2 + 0.22^2 + 0.08^2 \\
&= 0.3336
\end{aligned}$$

$$\begin{aligned}
Homogeneity &= \sum_{i,j} \frac{P_{45}(i,j)}{1+(i-j)^2} \\
&= \frac{0.48}{1+(0-0)^2} + \frac{0.22}{1+(0-1)^2} + \frac{0.22}{1+(1-0)^2} + \frac{0.08}{1+(1-1)^2} \\
&= 0.48 + 0.11 + 0.11 + 0.08 = 0.78
\end{aligned}$$

### 3) Filter Vertical (90°)

Tahapan dalam mencari filter horizontal GLCM adalah sebagai berikut :

a) Menentukan arah perhitungan GLCM dari matriks citra awal

$$\begin{pmatrix}
\begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{matrix} &
\begin{matrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} &
\begin{matrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}
\end{pmatrix}$$



b) Menentukan GLCM.

GLCM merupakan matriks  $2 \times 2$  dengan 2 *gray tone* yang nilainya 0 dan

1. Matriks ini diperoleh dari menghitung jumlah *gray tone*  $i$  dan  $j$  yang memiliki ketetanggaan yang sesuai.

<i>Gray tone</i>	0	1
0	(0,0)	(0,1)
1	(1,0)	(1,1)

Sehingga *gray tone* dari matriks citra asli didapatkan sebagai berikut :

<i>Gray tone</i>	0	1
0	17	4
1	5	3

Sehingga didapatkan :

$$\text{GLCM} = \begin{pmatrix} 17 & 4 \\ 5 & 3 \end{pmatrix}$$

c) Membuat GLCM simetri:

i. *Transpose* GLCM =  $\begin{pmatrix} 17 & 5 \\ 4 & 3 \end{pmatrix}$

ii. Jumlahkan GLCM dan *transpose* GLCM

$$\begin{pmatrix} 17 & 4 \\ 5 & 3 \end{pmatrix} + \begin{pmatrix} 17 & 5 \\ 4 & 3 \end{pmatrix} = \begin{pmatrix} 34 & 9 \\ 9 & 6 \end{pmatrix}$$

d) Menghitung total keseluruhan nilai matriks.

Total keseluruhan nilai matriks adalah 58

e) Menentukan matriks Normalisasi

Membagi setiap elemen matriks dengan total nilai keseluruhan matriks.

$$\text{Normalisasi} = \begin{pmatrix} \frac{34}{58} & \frac{9}{58} \\ \frac{9}{58} & \frac{6}{58} \end{pmatrix} = \begin{pmatrix} 0.586 & 0.155 \\ 0.155 & 0.103 \end{pmatrix}$$

$$P_{90}(i,j) = \begin{pmatrix} 0.586 & 0.155 \\ 0.155 & 0.103 \end{pmatrix}$$

f) Setelah didapatkan matriks  $P_{90}(i,j)$ , selanjutnya mencari nilai *contrast*, *correlation*, *energy* dan *homogeneity*.

$$\begin{aligned}
\text{Contrast} &= \sum_{i,j} (i-j)^2 P_{90}(i,j) \\
&= (0-0)^2 0.586 + (0-1)^2 0.155 + (1-0)^2 0.155 \\
&\quad + (1-1)^2 0.103 \\
&= 0.3125
\end{aligned}$$

$$\begin{aligned}
\mu_i &= \sum_{i,j} i P_{90}(i,j) \\
&= (0)0.586 + (1)0.155 + (0)0.155 + (1)0.103 = 0.258
\end{aligned}$$

$$\begin{aligned}
\mu_j &= \sum_{i,j} j P_{90}(i,j) \\
&= (0)0.586 + (1)0.155 + (0)0.155 + (1)0.103 \\
&= 0.258
\end{aligned}$$

$$\begin{aligned}
\sigma_i^2 &= \sum_{i,j} (i - \mu_i)^2 P_{90}(i,j) \\
&= (0 - 0.258)^2 0.586 + (1 - 0.258)^2 0.155 + (0 - 0.258)^2 0.155 \\
&\quad + (1 - 0.258)^2 0.103 \\
&= 0.1914
\end{aligned}$$

$$\begin{aligned}
\sigma_j^2 &= \sum_{i,j} (j - \mu_j)^2 P_{90}(i,j) \\
&= (0 - 0.258)^2 0.586 + (1 - 0.258)^2 0.155 + (0 - 0.258)^2 0.155 \\
&\quad + (1 - 0.258)^2 0.103 \\
&= 0.1914
\end{aligned}$$

$$\begin{aligned}
\text{Correlation} &= \sum_{i,j} P_{90}(i,j) \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \\
&= 0.586 \left[ \frac{(0-0.258)(0-0.258)}{\sqrt{(0.1914)(0.1914)}} \right] + 0.155 \left[ \frac{(1-0.258)(0-0.258)}{\sqrt{(0.1914)(0.1914)}} \right] \\
&\quad + 0.155 \left[ \frac{(0-0.258)(1-0.258)}{\sqrt{(0.1914)(0.1914)}} \right] + 0.103 \left[ \frac{(1-0.258)(1-0.258)}{\sqrt{(0.1914)(0.1914)}} \right] \\
&= 0.586(0.3478) + 0.155(-1.0002) + 0.155(-1.0002) \\
&\quad + 0.103(2.877) \\
&= 0.3233
\end{aligned}$$

$$\begin{aligned}
\text{Energy} &= \sum_{i,j} P_{90}(i,j)^2 \\
&= 0.586^2 + 0.155^2 + 0.155^2 + 0.103^2 \\
&= 0.4021
\end{aligned}$$

$$\begin{aligned}
Homogeneity &= \sum_{i,j} \frac{P_{90}(i,j)}{1+(i-j)^2} \\
&= \frac{0.586}{1+(0-0)^2} + \frac{0.155}{1+(0-1)^2} + \frac{0.155}{1+(1-0)^2} + \frac{0.103}{1+(1-1)^2} \\
&= 0.586 + 0.0775 + 0.0775 + 0.103 \\
&= 1.5415
\end{aligned}$$

#### 4) Filter Diagonal Kiri Atas ke Kanan Bawah (135°)

Tahapan dalam mencari filter horizontal GLCM adalah sebagai berikut:

a) Menentukan arah perhitungan GLCM dari matriks citra awal

$$\begin{pmatrix}
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}$$

b) Menentukan GLCM.

GLCM merupakan matriks  $2 \times 2$  dengan 2 *gray tone* yang nilainya 0 dan 1. Matriks ini diperoleh dari menghitung jumlah *gray tone*  $i$  dan  $j$  yang memiliki ketetanggaan yang sesuai.

<i>Gray tone</i>	0	1
0	(0,0)	(0,1)
1	(1,0)	(1,1)

Sehingga *gray tone* dari matriks citra asli didapatkan sebagai berikut:

<i>Gray tone</i>	0	1
0	13	4
1	6	2

Sehingga didapatkan :

$$GLCM = \begin{pmatrix} 13 & 4 \\ 6 & 2 \end{pmatrix}$$

c) Membuat GLCM simetri:

$$i. \text{ Transpose GLCM} = \begin{pmatrix} 13 & 6 \\ 4 & 2 \end{pmatrix}$$

ii. Jumlahkan GLCM dan *transpose* GLCM

$$\begin{pmatrix} 13 & 4 \\ 6 & 2 \end{pmatrix} + \begin{pmatrix} 13 & 6 \\ 4 & 2 \end{pmatrix} = \begin{pmatrix} 26 & 10 \\ 10 & 4 \end{pmatrix}$$

- d) Menghitung total keseluruhan nilai matriks.

Total keseluruhan nilai matriks adalah 50

- e) Menentukan matriks Normalisasi

Membagi setiap elemen matriks dengan total nilai keseluruhan matriks.

$$\text{Normalisasi} = \begin{pmatrix} \frac{26}{50} & \frac{10}{50} \\ \frac{10}{50} & \frac{4}{50} \end{pmatrix} = \begin{pmatrix} 0.52 & 0.2 \\ 0.2 & 0.08 \end{pmatrix}$$

$$P_{135}(i, j) = \begin{pmatrix} 0.52 & 0.2 \\ 0.2 & 0.08 \end{pmatrix}$$

- f) Setelah didapatkan matriks  $P_{135}(i, j)$ , selanjutnya mencari nilai *contrast*, *correlation*, *energy* dan *homogeneity*.

$$\begin{aligned} \text{Contrast} &= \sum_{i,j} (i - j)^2 P_{135}(i, j) \\ &= (0 - 0)^2 0.52 + (0 - 1)^2 0.2 + (1 - 0)^2 0.2 + (1 - 1)^2 0.08 \\ &= 0.4 \end{aligned}$$

$$\begin{aligned} \mu_i &= \sum_{i,j} i P_{135}(i, j) \\ &= (0)0.52 + (1)0.2 + (0)0.2 + (1)0.08 \\ &= 0.28 \end{aligned}$$

$$\begin{aligned} \mu_j &= \sum_{i,j} j P_{135}(i, j) \\ &= (0)0.52 + (1)0.2 + (0)0.2 + (1)0.08 \\ &= 0.28 \end{aligned}$$

$$\begin{aligned} \sigma_i^2 &= \sum_{i,j} (i - \mu_i)^2 P_{135}(i, j) \\ &= (0 - 0.28)^2 0.52 + (1 - 0.28)^2 0.2 + (0 - 0.28)^2 0.2 \\ &\quad + (1 - 0.28)^2 0.08 \\ &= 0.1452 \end{aligned}$$

$$\begin{aligned} \sigma_j^2 &= \sum_{i,j} (j - \mu_j)^2 P_{135}(i, j) \\ &= (0 - 0.28)^2 0.52 + (1 - 0.28)^2 0.2 + (0 - 0.28)^2 0.2 \\ &\quad + (1 - 0.28)^2 0.08 \\ &= 0.1452 \end{aligned}$$

$$\text{Correlation} = \sum_{i,j} P_{135}(i, j) \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right]$$

$$\begin{aligned}
&= 0.52 \left[ \frac{(0-0.28)(0-0.28)}{\sqrt{(0.1452)(0.1452)}} \right] + 0.2 \left[ \frac{(1-0.28)(0-0.28)}{\sqrt{(0.1452)(0.1452)}} \right] \\
&\quad + 0.2 \left[ \frac{(0-0.28)(1-0.28)}{\sqrt{(0.1452)(0.1452)}} \right] + 0.08 \left[ \frac{(1-0.28)(1-0.28)}{(0.1452)(0.1452)} \right] \\
&= 0.52(0.5399) + 0.2(-1.3884) + 0.2(-1.3884) \\
&\quad + 0.08(3.5702) \\
&= 0.011
\end{aligned}$$

$$\begin{aligned}
Energy &= \sum_{i,j} P_{135}(i,j)^2 \\
&= 0.52^2 + 0.2^2 + 0.2^2 + 0.08^2 \\
&= 0.3568
\end{aligned}$$

$$\begin{aligned}
Homogeneity &= \sum_{i,j} \frac{P_{135}(i,j)}{1+(i-j)^2} \\
&= \frac{0.52}{1+(0-0)^2} + \frac{0.2}{1+(0-1)^2} + \frac{0.2}{1+(1-0)^2} + \frac{0.08}{1+(1-1)^2} \\
&= 0.52 + 0.1 + 0.1 + 0.08 = 1.52
\end{aligned}$$

Sehingga fitur bentuk dari Gambar 4.15 adalah sebagai berikut:

Sudut 0°	<i>Contrast</i>	0.37
	<i>Correlation</i>	0.1662
	<i>Energy</i>	0.3208
	<i>Homogeneity</i>	0.813
Sudut 45°	<i>Contrast</i>	0.44
	<i>Correlation</i>	0.024
	<i>Energy</i>	0.3336
	<i>Homogeneity</i>	0.78
Sudut 90°	<i>Contrast</i>	0.3125
	<i>Correlation</i>	0.3233
	<i>Energy</i>	0.4021
	<i>Homogeneity</i>	1.54151
Sudut 135°	<i>Contrast</i>	0.4
	<i>Correlation</i>	0.011
	<i>Energy</i>	0.358
	<i>Homogeneity</i>	1.52

Nilai fitur tekstur dengan menggunakan GLCM untuk data citra tunggal disajikan pada lampiran 2.

#### 4.3 Klasifikasi dengan Menggunakan Metode SVM

Pada penelitian ini, data *input* yang digunakan berupa sebuah matriks yang merepresentasikan sebuah citra. Sedangkan data *input* untuk klasifikasi menggunakan metode *Support Vector Machine* berupa vektor. Sehingga dibutuhkan transformasi data dari matriks citra ke vektor dengan melakukan proses *preprocessing* dan ekstraksi fitur. Dengan kata lain, data masukan untuk klasifikasi dengan metode SVM berupa vektor yang berasal dari 22 nilai hasil atribut ekstraksi fitur yang terdiri dari 6 fitur bentuk dan 16 fitur tekstur. Dataset yang digunakan untuk data training berupa citra tunggal daun (1 citra terdiri dari 1 objek) sebanyak 189 data citra, sedangkan dataset yang digunakan untuk data testing berupa citra multi daun (1 citra terdiri atas lebih dari 1 objek) sebanyak 19 data citra.

Klasifikasi dengan metode SVM dibagi menjadi dua proses, yaitu proses *training* dan *testing*. Pada proses *training* SVM menggunakan data ekstraksi fitur citra daun. Sedangkan pada proses *testing* SVM menggunakan nilai bobot, bias dan alpha yang dihasilkan dari proses *training* yang kemudian dikalikan dengan data uji (sampel *testing*) sebagai *input*.

Algoritma proses *training* dapat dijabarkan sebagai berikut:

- a. Matriks  $X_{train}$  dan vektor  $Y$  sebagai pasangan *input* target dan *output* berupa  $w, \alpha, b$  yang merupakan variabel dari *hyperplane*. *Input* matriks  $X$  merupakan matriks yang berisi informasi data ekstraksi fitur dan vektor  $Y$  sebagai target. Oleh karena matriks  $X$  (data ekstraksi fitur citra daun) berjumlah 22 fitur dan jumlah data citra daun pada *training* adalah 189, maka matriks  $X$  berdimensi  $189 \times 22$ . Vektor  $Y$  merupakan vektor kolom untuk pengklasifikasi pertama dimana daun semangka disimbolkan dengan angka 1 dan daun gulma disimbolkan dengan angka -1.
- b. Langkah selanjutnya adalah menghitung matriks *Hessian*, yaitu perkalian antara kernel *quadratic* dengan  $Y$ . Matriks *Hessian* ini digunakan sebagai variabel *input* dalam *quadratic programming*. Fungsi *quadratic programming*

memerlukan variabel  $C$  sehingga perlu menetapkan nilai  $C$  ( $C$  adalah batas atas nilai  $\alpha_i$ ).

- c. Penyelesaian dengan *quadratic programming* adalah sebagai berikut:  
Selesaikan

$$\min L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j H$$

dimana

$$\sum_{i=1}^n \alpha_i y_i = 0 \text{ dan } 0 \leq \alpha \leq C.$$

dengan  $H = y_i y_j K(X_i, X_j)$  dan  $K(X_i, X_j) = \varphi(X_i) \cdot \varphi(X_j)$  adalah fungsi kernelnya, dan  $e$  adalah vektor satuan yang memiliki dimensi yang sama dengan  $Y$  dengan  $C > 0$  adalah batas atas dari nilai  $\alpha$ .

Hasil dari fungsi *quadratic programming* adalah nilai variabel  $w, \alpha$  dan  $b$  untuk masing-masing kelas yang digunakan untuk proses uji.

Implementasi dari metode SVM menggunakan 4 data matriks citra yang yang telah ditransformasi dengan melakukan *preprocessing* dan ekstraksi fitur dimana 2 data mewakili semangka dan 2 data lainnya mewakili gulma. Atribut data set yang digunakan adalah 10 informasi ekstraksi fitur yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Contoh Data yang Menggunakan Metode *Support Vector Machine*

No	AR	FF	R	FN	RPD	RPPLF	Cont	Corr	En	Hom	Kelas
1.	0.391	16.242	4.706	0.077	0.074	0.077	0.077	0.126	0.84	0.961	1
2.	0.454	12.542	4.135	0.665	0.062	0.665	0.665	0.131	0.864	0.967	1
3.	0.74	9.819	3.850	0.075	0.075	0.076	0.075	0.157	0.841	0.962	-1
4.	0.667	10.820	3.983	0.060	0.057	0.060	0.060	0.145	0.873	0.97	-1

Keterangan :

AR = Aspek Rasio

FF = Faktor *Form*

R = *Rectangularity*

FN = Faktor *Narrow*

RPD = Rasio Perimeter dari Diameter

RPPLF= Rasio Perimeter dari Panjang dan Lebar Fisiologis

Cont = *Contrast*

Cor = *Correlation*

En = *Energy*

Hom = *Homogeneity*



Dari data tersebut kita mempunyai data :

$$X_1 = [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961]$$

$$X_2 = [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967]$$

$$X_3 = [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962]$$

$$X_4 = [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97]$$

Selanjutnya melakukan kernelisasi sehingga didapatkan fungsi *quadratic* di dalam *feature space*. Pada permasalahan ini, digunakan kernel *Polynomial* derajat 2 yang didefinisikan sebagai berikut:

$$K(X_i, X_j) = (X_i \cdot X_j^T + 1)^2$$

Formulasi yang digunakan adalah *dualitas Lagrange multiplier* pada persamaan (2.25) sehingga untuk mendapatkan  $w$  menggunakan persamaan tersebut yang sudah dimodifikasi dengan fungsi kernelnya menjadi

$$w = \sum_{i=1}^n \alpha_i y_i \varphi(X)$$

Kernel yang dipilih berderajat dua agar nilai hasil kernel tidak begitu besar, sehingga dapat dihitung matriks kernel dengan dimensi  $l \times l$ , dimana  $l$  adalah banyaknya data *training*. *Dot product* dari  $K(X_i, X_j)$  dapat dihitung sebagai berikut:

$$\begin{aligned} X_1 \cdot X_1^T &= [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \cdot \\ &\quad [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \\ &= 287.776 \end{aligned}$$

$$(X_1 \cdot X_1^T + 1)^2 = (287.776 + 1)^2 = 83391.49$$

$$\begin{aligned} X_1 \cdot X_2^T &= [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \cdot \\ &\quad [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \\ &= 225.038 \end{aligned}$$

$$(X_1 \cdot X_2^T + 1)^2 = (225.038 + 1)^2 = 51093.341$$

$$\begin{aligned}
X_1 \cdot X_3^T &= [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \cdot \\
&\quad [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \\
&= 179.569
\end{aligned}$$

$$(X_1 \cdot X_3^T + 1)^2 = (179.569 + 1)^2 = 32605.059$$

$$\begin{aligned}
X_1 \cdot X_4^T &= [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \cdot \\
&\quad [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \\
&= 196.452
\end{aligned}$$

$$(X_1 \cdot X_4^T + 1)^2 = (196.452 + 1)^2 = 38987.209$$

$$\begin{aligned}
X_2 \cdot X_1^T &= [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \cdot \\
&\quad [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \\
&= 225.038
\end{aligned}$$

$$(X_2 \cdot X_1^T + 1)^2 = (225.038 + 1)^2 = 51093.341$$

$$\begin{aligned}
X_2 \cdot X_2^T &= [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \cdot \\
&\quad [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \\
&= 176.324
\end{aligned}$$

$$(X_2 \cdot X_2^T + 1)^2 = (176.324 + 1)^2 = 31443.713$$

$$\begin{aligned}
X_2 \cdot X_3^T &= [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \cdot \\
&\quad [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \\
&= 141.108
\end{aligned}$$

$$(X_2 \cdot X_3^T + 1)^2 = (141.108 + 1)^2 = 20194.756$$

$$\begin{aligned}
X_2 \cdot X_4^T &= [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \cdot \\
&\quad [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \\
&= 154.209
\end{aligned}$$

$$(X_2 \cdot X_4^T + 1)^2 = (154.209 + 1)^2 = 24089.699$$

$$\begin{aligned}
X_3 \cdot X_1^T &= [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \cdot \\
&\quad [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \\
&= 179.569
\end{aligned}$$

$$(X_3 \cdot X_1^T + 1)^2 = (179.569 + 1)^2 = 32605.059$$

$$\begin{aligned}
X_3 \cdot X_2^T &= [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \cdot \\
&\quad [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \\
&= 141.108
\end{aligned}$$

$$(X_3 \cdot X_2^T + 1)^2 = (141.108 + 1)^2 = 20194.756$$

$$\begin{aligned}
X_3 \cdot X_3^T &= [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \cdot \\
&\quad [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \\
&= 113.471
\end{aligned}$$

$$(X_3 \cdot X_3^T + 1)^2 = (113.471 + 1)^2 = 13103.502$$

$$\begin{aligned}
X_3 \cdot X_4^T &= [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \cdot \\
&\quad [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \\
&= 123.785
\end{aligned}$$

$$(X_3 \cdot X_4^T + 1)^2 = (123.785 + 1)^2 = 15571.3$$

$$\begin{aligned}
X_4 \cdot X_1^T &= [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \cdot \\
&\quad [0.391, 16.242, 4.706, 0.077, 0.074, 0.077, 0.077, 0.126, 0.84, 0.961] \\
&= 196.452
\end{aligned}$$

$$(X_4 \cdot X_1^T + 1)^2 = (196.452 + 1)^2 = 38987.209$$

$$\begin{aligned}
X_4 \cdot X_2^T &= [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \cdot \\
&\quad [0.454, 12.542, 4.135, 0.665, 0.062, 0.665, 0.665, 0.131, 0.864, 0.967] \\
&= 154.209
\end{aligned}$$

$$(X_4 \cdot X_2^T + 1)^2 = (154.209 + 1)^2 = 24089.699$$

$$\begin{aligned}
X_4 \cdot X_3^T &= [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \cdot \\
&\quad [0.74, 9.819, 3.850, 0.075, 0.075, 0.076, 0.075, 0.157, 0.841, 0.962] \\
&= 123.785
\end{aligned}$$

$$(X_4 \cdot X_3^T + 1)^2 = (123.785 + 1)^2 = 15571.3$$

$$\begin{aligned}
X_4 \cdot X_4^T &= [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \cdot \\
&\quad [0.667, 10.820, 3.983, 0.060, 0.057, 0.060, 0.060, 0.145, 0.873, 0.97] \\
&= 135.127
\end{aligned}$$

$$(X_4 \cdot X_4^T + 1)^2 = (135.127 + 1)^2 = 18530.469$$

Sehingga didapatkan matriks  $K$  sebagai berikut:

$$K = \begin{bmatrix} X_1 \cdot X_1^T & X_1 \cdot X_2^T & X_1 \cdot X_3^T & X_1 \cdot X_4^T \\ X_2 \cdot X_1^T & X_2 \cdot X_2^T & X_2 \cdot X_3^T & X_2 \cdot X_4^T \\ X_3 \cdot X_1^T & X_3 \cdot X_2^T & X_3 \cdot X_3^T & X_3 \cdot X_4^T \\ X_4 \cdot X_1^T & X_4 \cdot X_2^T & X_4 \cdot X_3^T & X_4 \cdot X_4^T \end{bmatrix}$$

$$K = \begin{bmatrix} 83391.490 & 51093.341 & 32605.059 & 38987.209 \\ 51093.341 & 31443.713 & 20194.756 & 24089.699 \\ 32605.059 & 20194.756 & 13103.502 & 15571.300 \\ 38987.209 & 24089.699 & 15571.300 & 18530.469 \end{bmatrix}$$

Matrks  $K$  setiap elemennya merupakan hasil kali  $X_i \cdot X_j^T$  yang berkorelasi dengan  $\alpha_i \alpha_j$  dalam persamaan (2.27). Dengan menggunakan matriks kernel  $K$  sebagai pengganti *dot product*  $X_i \cdot X_j^T$  dalam persamaan *dualitas Lagrange multiplier* (2.27), di dapatkan :

$$\begin{aligned} L_D &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (X_i \cdot X_j^T) \\ &= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2} ((83391.490)(1)(1)\alpha_1\alpha_1) \\ &\quad + ((51093.341)(1)(1)\alpha_1\alpha_2) + ((32605.059)(1)(-1)\alpha_1\alpha_3) \\ &\quad + ((38987.209)(1)(-1)\alpha_1\alpha_4) + ((51093.341)(1)(1)\alpha_2\alpha_1) \\ &\quad + ((31443.713)(1)(1)\alpha_2\alpha_2) + ((20194.756)(1)(-1)\alpha_2\alpha_3) \\ &\quad + ((24089.699)(1)(-1)\alpha_2\alpha_4) + ((32605.059)(-1)(1)\alpha_3\alpha_1) \\ &\quad + ((20194.756)(-1)(1)\alpha_3\alpha_2) + ((13103.502)(-1)(-1)\alpha_3\alpha_3) \\ &\quad + ((15571.300)(-1)(-1)\alpha_3\alpha_4) + ((38987.209)(-1)(1)\alpha_4\alpha_1) \\ &\quad + ((24089.699)(-1)(1)\alpha_4\alpha_2) + ((15571.300)(-1)(-1)\alpha_4\alpha_3) \\ &\quad + ((18530.469)(-1)(-1)\alpha_4\alpha_4)) \\ &= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - \frac{1}{2} (83391.490\alpha_1\alpha_1 + 51093.341\alpha_1\alpha_2 \\ &\quad - 32605.059\alpha_1\alpha_3 - 38987.209\alpha_1\alpha_4 + 51093.341\alpha_2\alpha_1 \\ &\quad + 31443.713\alpha_2\alpha_2 - 20194.756\alpha_2\alpha_3 - 24089.699\alpha_2\alpha_4 \\ &\quad - 32605.059\alpha_3\alpha_1 - 20194.756\alpha_3\alpha_2 + 13103.502\alpha_3\alpha_3 \\ &\quad + 15571.300\alpha_3\alpha_4 - 38987.209\alpha_4\alpha_1 - 24089.699\alpha_4\alpha_2 \\ &\quad + 15571.300\alpha_4\alpha_3 + 18530.469\alpha_4\alpha_4) \\ &= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) - (\frac{1}{2} \cdot 83391.490\alpha_1^2 + 51093.341\alpha_1\alpha_2 \\ &\quad - 32605.059\alpha_1\alpha_3 - 38987.209\alpha_1\alpha_4 + \frac{1}{2} \cdot 31443.713\alpha_2^2 \end{aligned}$$

$$-20194.756\alpha_2\alpha_3 - 24089.699\alpha_2\alpha_4 + \frac{1}{2} \cdot 13103.502\alpha_3^2 \\ + 15571.300\alpha_3\alpha_4 + \frac{1}{2} \cdot 18530.469\alpha_4^2)$$

Syarat 1:

$$\sum_{i=1}^n \alpha_i y_i = 0$$

$$\alpha_1 y_1 + \alpha_2 y_2 + \alpha_3 y_3 + \alpha_4 y_4 = 0$$

$$\alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 = 0$$

Syarat 2:

$$0 \leq \alpha_1, \alpha_2, \alpha_3, \alpha_4 \leq C$$

Penyelesaian :

$$\frac{dL_D}{d\alpha_1} = 0 :$$

$$1 - 83391.490 \alpha_1 - 51093.341 \alpha_2 + 32605.059 \alpha_3 + 38987.209 \alpha_4 = 0$$

$$\frac{dL_D}{d\alpha_2} = 0 :$$

$$1 - 51093.341 \alpha_1 - 31443.713 \alpha_2 + 20194.756 \alpha_3 - 24089.699 \alpha_4 = 0$$

$$\frac{dL_D}{d\alpha_3} = 0 :$$

$$1 + 32605.059 \alpha_1 - 20194.756 \alpha_2 + 13103.502 \alpha_3 + 15571.300 \alpha_4 = 0$$

$$\frac{dL_D}{d\alpha_4} = 0 :$$

$$1 - 38987.209 \alpha_1 - 24089.699 \alpha_2 + 15571.300 \alpha_3 + 18530.469 \alpha_4 = 0$$

(i) Menghilangkan nilai  $\alpha_1$

$$\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_2} :$$

$$0.0032 \cdot 10^7 - 1.1609 \cdot 10^7 \alpha_2 + 1.8169 \cdot 10^7 \alpha_3 + 1.6889 \cdot 10^7 \alpha_4 = 0$$

$$\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_3} :$$

$$0.0116 \cdot 10^7 + 1.8169 \cdot 10^7 \alpha_2 - 2.9631 \cdot 10^7 \alpha_3 - 2.7334 \cdot 10^7 \alpha_4 = 0$$

$$\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_4} :$$

$$0.0122 \cdot 10^7 + 1.6889 \cdot 10^7 \alpha_2 - 2.7334 \cdot 10^7 \alpha_3 - 2.5281 \cdot 10^7 \alpha_4 = 0$$

(ii) Menghilangkan nilai  $\alpha_2$

$$\left( \frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_2} \right) - \left( \frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_3} \right) :$$

$$0.1933 \cdot 10^{13} - 1.3843 \cdot 10^{13} \alpha_3 - 1.0440 \cdot 10^{13} \alpha_4 = 0$$

$$\left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_2}\right) - \left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_4}\right) :$$

$$0.1966 \cdot 10^{13} - 1.0440 \cdot 10^{13} \alpha_3 - 0.8233 \cdot 10^{13} \alpha_4 = 0$$

(iii) Mendapatkan nilai  $\alpha_4$

$$\left(\left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_2}\right) - \left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_3}\right)\right) - \left(\left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_2}\right) - \left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_4}\right)\right) :$$

$$7.0348 \cdot 10^{24} - 4.9758 \cdot 10^{24} \alpha_4 = 0$$

$$-4.9758 \cdot 10^{24} \alpha_4 = -7.0348 \cdot 10^{24}$$

$$\alpha_4 = 1.414$$

(iv) Substitusi nilai  $\alpha_4$  ke Persamaan  $\left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_2}\right) - \left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_3}\right) :$

$$0.1933 \cdot 10^{13} - 1.3843 \cdot 10^{13} \alpha_3 - 1.0440 \cdot 10^{13} \alpha_4 = 0$$

$$0.1933 \cdot 10^{13} - 1.3843 \cdot 10^{13} \alpha_3 - 1.0440 \cdot 10^{13} \cdot 1.414 = 0$$

$$-1.3843 \cdot 10^{13} \alpha_3 = 1.2829 \cdot 10^{13}$$

$$\alpha_3 = -0.9267$$

(v) Substitusi nilai  $\alpha_3$  dan  $\alpha_4$  ke Persamaan  $\left(\frac{dL_D}{d\alpha_1} - \frac{dL_D}{d\alpha_3}\right) :$

$$0.0116 \cdot 10^7 + 1.8169 \cdot 10^7 \alpha_2 - 2.9631 \cdot 10^7 \alpha_3 - 2.7334 \cdot 10^7 \alpha_4 = 0$$

$$0.0116 \cdot 10^7 + 1.8169 \cdot 10^7 \alpha_2 - 2.9631 \cdot 10^7 \cdot (-0.9267) - 2.7334 \cdot 10^7 \cdot 1.414 = 0$$

$$1.8169 \cdot 10^7 \alpha_2 = 1.1084$$

$$\alpha_2 = 0.6101$$

(vi) Substitusi nilai  $\alpha_2$ ,  $\alpha_3$  dan  $\alpha_4$  ke Persamaan  $\frac{dL_D}{d\alpha_1} :$

$$1 - 83391.490 \alpha_1 - 51093.341 \alpha_2 + 32605.059 \alpha_3 + 38987.209 \alpha_4 = 0$$

$$1 - 83391.490 \alpha_1 - 51093.341 \cdot 0.6101 + 32605.059 \cdot (-0.9267) +$$

$$38987.209 \cdot 1.414 = 0$$

$$-83391.490 \alpha_1 = 6258.2419$$

$$\alpha_1 = -0.0751$$

Karena syarat 2 adalah  $0 \leq \alpha_1, \alpha_2, \alpha_3, \alpha_4 \leq C$  maka  $\alpha_2$  dan  $\alpha_4$  adalah *support vector*.



Dalam fungsi tujuan, karena suku kedua sudah dikalikan dengan  $y_i y_j$ , maka persamaan tersebut memenuhi standar *Quadratic Programming* (QP) sehingga menghasilkan nilai  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ . Nilai  $\alpha_2, \alpha_4$  yang telah diperoleh digunakan untuk menentukan nilai  $w$  dan  $b$ . Nilai  $w$  sebenarnya tidak dapat ditampilkan secara eksplisit karena merupakan gabungan dari  $\alpha, y$ , dan  $\varphi(X)$ . Sedangkan dalam banyak hal  $\varphi(X)$  sulit dihitung, kecuali kernel linier di mana  $\varphi(X) = x$ . Nilai  $b$  diperoleh dari proses pelatihan yang dilakukan.

Setelah mendapatkan nilai  $\alpha$ , dan  $b$ , metode Support Vector Machine dapat digunakan untuk klasifikasi. Maksimum Marginal Hyperplane yang digunakan adalah :

$$f(x_d) = \text{sign} \left( \sum_{i=1}^l \alpha_i y_i \varphi(X_i) \cdot \varphi(X_z) + b \right)$$

karena kernel yang digunakan adalah kernel polinomial derajat 2 maka  $f(x_d)$  menjadi :

$$f_1 = \text{sign} \left( \sum_{i=1}^l \alpha_i y_i K(X_i \cdot X_z + 1)^2 + b \right)$$

Keterangan:

$\varphi(X_i) \cdot \varphi(X_z)$  = *inner product* antara data *support vector* dengan data uji.

$\text{sign}$  = fungsi *signum* yaitu nilai yang lebih besar dari 0 dilabelkan +1 dan semua nilai yang lebih kecil dari 0 dilabelkan -1.

$X_z$  =  $(z_{11}, z_{12}, \dots, z_{1k})$ ,  $k$  adalah jumlah data uji.

#### 4.4 Implementasi Proses Klasifikasi dengan Matlab

Implementasi proses klasifikasi dengan matlab dibagi menjadi tiga tahap yaitu implementasi *preprocessing*, implementasi ekstraksi fitur, dan implementasi metode *Support Vector Machine* (SVM).

##### 4.4.1 Implementasi *Preprocessing*

Implementasi proses *preprocessing* dengan matlab dilakukan dalam dua proses. Proses pertama adalah pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan, dilanjutkan dengan proses kedua yaitu segmentasi deteksi tepi dengan operator *Canny*.

**a. Pemisahan *Background* dan *Foreground* serta Menghilangkan Objek yang Tidak Diinginkan**

Pada tahap pemisahan *background* data citra yang digunakan adalah citra yang telah melalui proses *cropping* dan *resize*. Citra yang telah di-*cropping* dan *resize* masih berupa citra *Red Green Blue* (RGB). Untuk pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan, ruang citra RGB dikonversi ke ruang citra *Hue Saturation Value* (HSV). Setelah itu diambil komponen intensitas H dan S dari HSV kemudian diubah ke bentuk citra biner (hitam putih).

Oleh karena warna daun pada tahap ini adalah hitam sedangkan warna daun yang diinginkan adalah berwarna putih, maka selanjutnya mengubah warna hitam menjadi warna putih (dinegasikan), dan sebaliknya. Lalu diberikan bingkai pada piksel baris pertama sampai ketiga dan kolom pertama sampai ketiga dengan menjadikannya bernilai 0. Selanjutnya, menggabungkan kedua citra tersebut dengan logika AND (“&”).

Tahap terakhir dari proses ini adalah menghilangkan objek yang tidak diinginkan dengan cara melakukan morfologi *opening* pada citra biner tersebut dengan struktur elemen (SE). Argumen SE menjadi struktur elemen objek tunggal, sebagai lawan array objek. Operasi morfologi *opening* adalah erosi yang diikuti oleh dilasi, menggunakan struktur elemen yang sama untuk kedua operasi. Struktur elemen yang digunakan berbentuk *disk* dengan radius 1 dan 4 ketetanggan. Hal ini memungkinkan untuk menghapus objek dengan ukuran yang lebih kecil dari struktur elemen. Proses ini diberi nama fungsi *mask*.

```

function mask = get_mask(Irgb)

Ihsv = rgb2hsv(Irgb);
Ih = imadjust(Ihsv(:,:,1));
Is = imadjust(Ihsv(:,:,2));

mask1 = im2bw(Ih, graythresh(Ih));
mask2 = im2bw(Is, graythresh(Is));

[m,n] = size(mask1);
if mask1(1,1) || mask1(1,n) || mask1(m,1) || mask1(m,n), mask1 =
~mask1; end
if mask2(1,1) || mask2(1,n) || mask2(m,1) || mask2(m,n), mask2 =
~mask2; end

mask1([1,2,3,m-2,m-1,m],:) = 0; mask1(:, [1,2,3,n-2,n-1,n]) = 0;
mask2([1,2,3,m-2,m-1,m],:) = 0; mask2(:, [1,2,3,n-2,n-1,n]) = 0;

mask = mask1 & mask2;
mask = imopen(mask, strel('disk',1,4));
end

```

#### b. Segmentasi Deteksi Tepi dengan Operator *Canny*

Deteksi tepi dengan operator *Canny* dilakukan menggunakan data citra hasil kali antara citra hasil pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan (fungsi *mask*) dengan citra asli yang diubah menjadi citra *grayscale*. Proses ini diberi nama fungsi tepi.

```

function [tepi, roi] = get_tepi(Irgb, mask)
roi = double(rgb2gray(Irgb));
roi = roi .* mask;
tepi = edge(roi, 'canny');
end

```

#### 4.4.2 Implementasi Ekstraksi Fitur

Fungsi *mask* dan tepi digunakan untuk mencari ekstraksi fitur daun. Fungsi *mask* digunakan untuk mencari ekstraksi fitur bentuk daun. Nilai 5 fitur dasar geometri (diameter, panjang fisiologis, lebar fisiologis, area, dan perimeter) diperoleh dengan menggunakan *toolbox* pada matlab.

Selanjutnya nilai 5 fitur geometri tersebut digunakan untuk mencari aspek rasio, faktor *form*, *rectangularity*, faktor *narrow*, rasio perimeter dari diameter serta rasio perimeter dari panjang fisiologis dan lebar fisiologis. Fungsi tepi digunakan untuk mencari ekstraksi fitur tekstur daun. *Contrast*, *correlation*,

*energy* dan *homogeneity* diperoleh menggunakan *toolbox* pada matlab dengan jarak 1, level 2 dan 4 sudut orientasi. Hasil dari ekstraksi fitur ini kemudian digunakan sebagai atribut dalam klasifikasi dengan metode *Support Vector Machine* (SVM).

```
function var_out = ekstraksi_fitur(mask, tepi)

%% fitur dasar geometri
stats = regionprops(mask,
{'EquivDiameter', 'MajorAxisLength', 'MinorAxisLength', 'Area',
'Perimeter'});
[~,idx] = max([stats.Area]);
diameter = stats(idx).EquivDiameter;
panjang_fisiologis = stats(idx).MajorAxisLength;
lebar_fisiologis = stats(idx).MinorAxisLength;
area = stats(idx).Area;
perimeter = stats(idx).Perimeter;
var_out(1) = panjang_fisiologis/lebar_fisiologis; % aspek
rasio
var_out(2) = (4*pi*area)/perimeter^2; % faktor form
var_out(3) = panjang_fisiologis*lebar_fisiologis/area; %
rectangularity
var_out(4) = diameter/panjang_fisiologis; % faktor narrow
var_out(5) = perimeter/diameter; % rasio perimeter dari
diameter
var_out(6) = perimeter/(panjang_fisiologis+lebar_fisiologis);
% rasio perimeter dari panjang dan lebar fisiologi

%% glcm
glcm = graycomatrix(tepi, 'Offset', [2 0; 0 2; -2 0; 0 -2]);
stats = graycoprops(glcm,
{'Contrast', 'Correlation', 'Energy', 'Homogeneity'});
var_out(7) = stats.Contrast(1);
var_out(8) = stats.Contrast(2);
var_out(9) = stats.Contrast(3);
var_out(10) = stats.Contrast(4);
var_out(11) = stats.Correlation(1);
var_out(12) = stats.Correlation(2);
var_out(13) = stats.Correlation(3);
var_out(14) = stats.Correlation(4);
var_out(15) = stats.Energy(1);
var_out(16) = stats.Energy(2);
var_out(17) = stats.Energy(3);
var_out(18) = stats.Energy(4);
var_out(19) = stats.Homogeneity(1);
var_out(20) = stats.Homogeneity(2);
var_out(21) = stats.Homogeneity(3);
var_out(22) = stats.Homogeneity(4);
end
```

#### 4.4.3 Implementasi Metode *Support Vector Machine* (SVM)

Matlab menyediakan fungsi untuk melakukan training dan prediksi dengan metode SVM, yaitu `svmtrain()` dan `svmclassify()`. Terdapat 22 fitur yang terdiri dari 6 fitur bentuk dan 4 fitur tekstur dengan 4 arah sehingga menjadi 16 fitur tekstur.

##### a. *Training*

Fungsi `svmtrain()` digunakan untuk melakukan pelatihan pada matriks data *training* dan kelas dalam matriks *group*.

```
function pushbutton1_Callback(hObject, eventdata, handles)

n_fitur = 22;
list_semangka = dir([handles.f_semangka, '*.jpg']);
n_semangka = length(list_semangka);
list_gulma = dir([handles.f_gulma, '*.jpg']);
n_gulma = length(list_gulma);

xdata = zeros(n_semangka+n_gulma, n_fitur);
kelas = zeros(n_semangka+n_gulma,1);

for i=1:n_semangka
    Irgb = imread([handles.f_semangka, list_semangka(i).name]);
    mask = get_mask(Irgb);
    tepi = get_tepi(Irgb, mask);
    xdata(i,:) = ekstraksi_fitur(mask, tepi);
    kelas(i) = 1;
end
for i=1:n_gulma
    Irgb = imread([handles.f_gulma, list_gulma(i).name]);
    mask = get_mask(Irgb);
    tepi = get_tepi(Irgb, mask);
    xdata(n_semangka+i,:) = ekstraksi_fitur(mask, tepi);
    kelas(n_semangka+i) = -1;
end

handles.svmStruct = svmtrain(xdata, kelas, 'kernel_function',
    'quadratic','method','QP');
```

Perintah pelatihan svm yaitu:

```
handles.svmStruct = svmtrain(Training, Groups,
    'kernel_function', 'quadratic', 'options', options);
handles.svmStruct = svmtrain(xdata, kelas, 'kernel_function',
    'quadratic','method','QP');
```

Keterangan :

- 1) svmStruct merupakan variabel bertipe struktur hasil pelatihan, yang menyimpan field-field sebagai berikut:
  - a) SupportVectors, data yang ikut menjadi *support vector*.
  - b) Alpha. Nilai Lagrange *multiplier* ( $\alpha$ ) yang didapatkan.
  - c) Bias, nilai bias ( $b$ ) yang didapatkan.
  - d) KernelFunction, nama fungsi kernel yang digunakan.
  - e) KernelFunctionArgs, nilai parameter kernel yang dipakai.
  - f) GroupsNames, label kelas data latih
  - g) SupportVectorsIndicates, indeks (nomor baris) data latih yang menjadi *support vector*.
  - h) ScaleData, nilai skala data.
  - i) FigureHandles.
- 2) Training merupakan perintah untuk data latih yang digunakan. Matriks  $M \times N$  adalah data latih dengan M adalah jumlah fitur, N adalah jumlah data latih.
- 3) Group adalah matriks  $1 \times N$  untuk kelas data latih yang digunakan. N adalah jumlah data. N(kolom) antara matriks *training* dan group harus sama.

## **b. Pengambilan Objek**

Data uji dalam penelitian ini adalah citra multi daun oleh karena itu dibutuhkan proses pengambilan objek agar bisa diklasifikasikan. Teknik *labeling* yaitu 'bwlabel' digunakan untuk melabeli semua komponen terhubung dalam citra biner (hitam putih) dan mengembalikannya. Citra biner yang digunakan adalah citra hasil pemisahan *background* dan *foreground* serta menghilangkan objek yang tidak diinginkan (fungsi *mask*). Jumlah objek yang ditemukan dalam citra dinilai *outputnya*. Label semua area terisolasi mempunyai nomor yang berbeda.

Operasi *opening* area digunakan untuk menghapus objek yang tidak diinginkan. Hal ini memungkinkan untuk menghapus objek dengan ukuran yang lebih kecil. Objek yang digunakan dalam data uji ini adalah yang jumlah pikselnya lebih dari 2000 piksel.

```

Irgb = imread([handles.f_tes, handles.list_citra_tes{val}]);
mask = get_mask(Irgb);

L = bwlabel(mask);
stats = regionprops(mask, 'Area');
areax = [stats.Area];
idx = find(areax > 2000);
mask = ismember(L, idx);

```

### c. Uji (*Testing*)

Data uji yang diklasifikasi dalam satu citra multi daun adalah objek yang telah ditentukan yaitu yang jumlah pikselnya lebih besar dari 2000 piksel. Fungsi `svmclassify()` digunakan untuk melakukan prediksi data uji dengan model yang sudah dibangun menggunakan `svmtrain()`.

```

function pushbutton2_Callback(hObject, eventdata, handles)

    n_semangka = 0;
    n_gulma = 0;
    L = bwlabel(handles.mask);
    stat1 = regionprops(handles.mask, 'Centroid', 'BoundingBox');
    l_max = max(max(L));
    h = waitbar(0/l_max, 'Pengenalan...');
    for i=1:l_max
        mask = L==i;
        tepi = handles.tepi .* mask;
        xdata = ekstraksi_fitur(mask, tepi);
        kelas = svmclassify(handles.svmStruct, xdata);
        if kelas == 1
            n_semangka = n_semangka + 1;
        else
            n_gulma = n_gulma + 1;
        end
    end
end

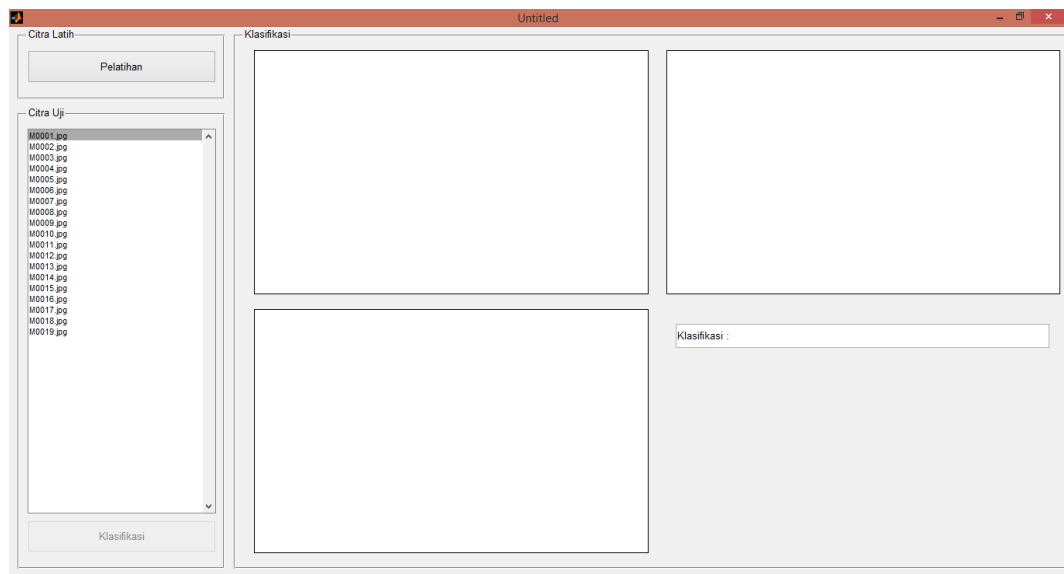
```

## 4.5 Uji Coba Program

Setelah dilakukan implementasi program, tahap berikutnya adalah uji coba program klasifikasi dengan metode SVM. Hasil yang diperoleh adalah sebagai berikut:

### a. Tampilan awal GUI

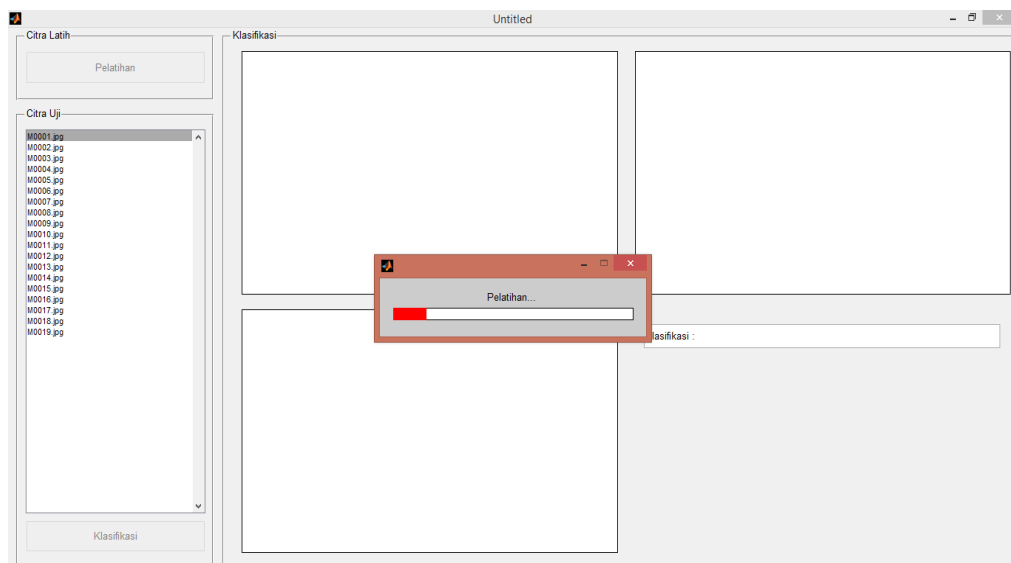
Hasil *running* GUI pada matlab ditampilkan pada Gambar 4.16. Data citra uji yang digunakan sebanyak 19 data citra.



Gambar 4.16 Tampilan Awal GUI

b. Proses *training*

Data yang digunakan pada proses *training* sebanyak 189 data citra yang terdiri dari 156 data citra daun semangka dan 33 data citra gulma. Sebelum dilakukan proses *training*, data citra terlebih dahulu mengalami proses *preprocessing* citra. Selanjutnya dilakukan ekstraksi fitur dan klasifikasi dengan menggunakan metode SVM .

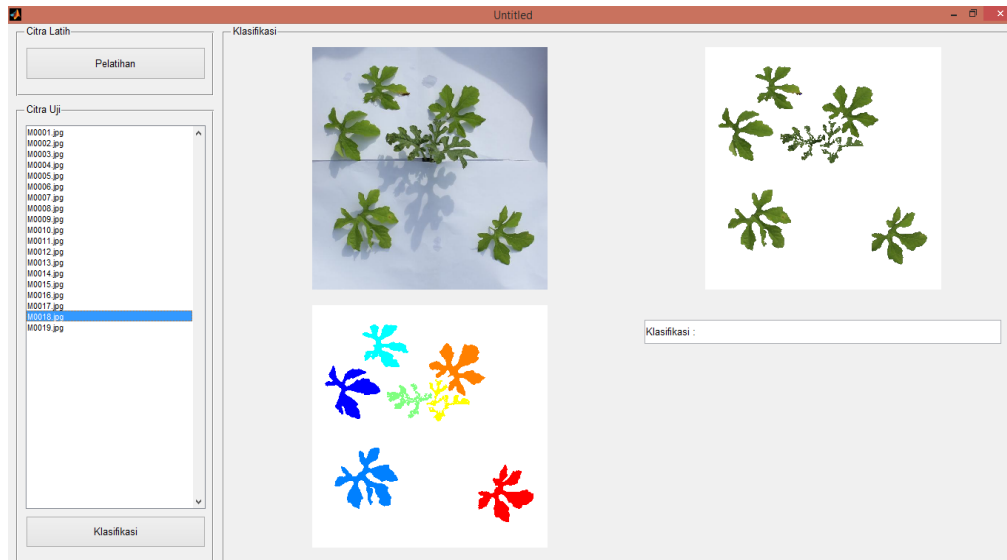


Gambar 4.17 Proses *Training*



c. Pemilihan data uji yang

Setelah data *training* dilakukan, diperoleh nilai  $b, w$  dan  $\alpha$  yang digunakan untuk proses *testing* data daun dengan multi daun.



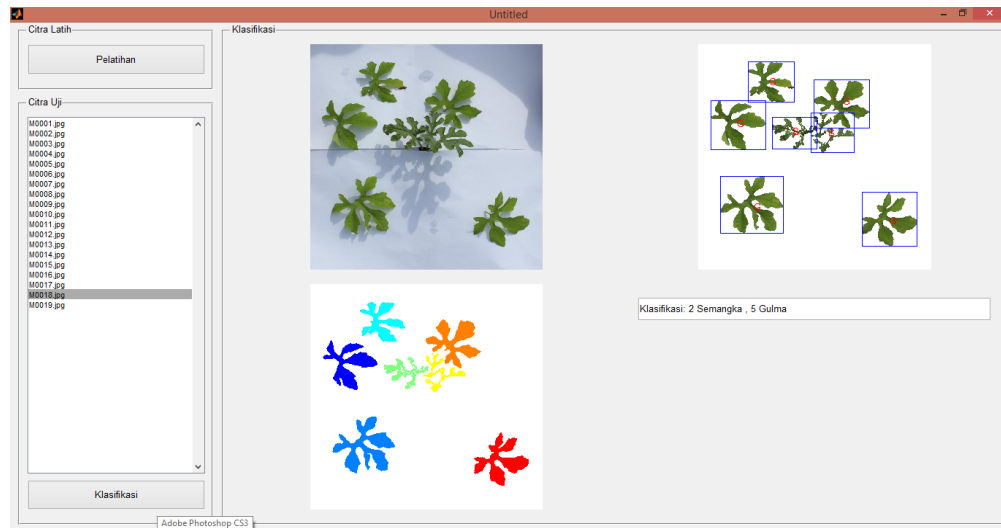
Gambar 4.18 Proses Pemilihan Data Uji

Pada Gambar 4.18 citra multi daun sebelah kiri atas menampilkan citra asli, citra multi daun sebelah kiri bawah menunjukkan hasil deteksi citra daun. Setiap warna yang berbeda menunjukkan daun yang berbeda. Bila terjadi kesalahan mendeteksi citra daun karena adanya tumpang tindih, maka daun yang terdeteksi tersebut tidak diikutsertakan dalam pengklasifikasian.

d. Proses klasifikasi citra multi daun

Proses klasifikasi citra multi daun dengan metode SVM sama dengan saat proses *training*. Klasifikasi citra multi daun dilakukan dengan mendeteksi daun kemudian diklasifikasikan dengan memperhatikan klasifikasi citra tunggal dari hasil *training*. Langkah awal dalam proses klasifikasi citra multi daun yaitu mendeteksi daun, setelah proses *testing* dilakukan satu-persatu dengan menggunakan 189 data citra tunggal yang telah diklasifikasikan pada saat *training*.

Citra daun sebelah kanan atas pada Gambar 4.19 menunjukkan daun yang diklasifikasikan dengan memberi *rectangle* pada setiap daun yang terdeteksi, kemudian memberi kode S bila daun tersebut dikenali dengan daun semangka dan kode G bila daun tersebut dikenali dengan gulma.



Gambar 4.19 Proses Klasifikasi Citra Multi Daun

Klasifikasi dari hasil data uji yang ditampilkan menunjukkan 2 daun semangka dan 5 daun gulma. Hasil dari data uji yang ditampilkan ini memiliki akurasi 100%.

#### 4.6 Analisis Hasil

Data *training* yang digunakan sebanyak 189 data citra yang terdiri dari 156 data citra daun semangka dan 33 data citra gulma. Sedangkan data uji yang digunakan adalah citra multi daun sebanyak 19 data citra. Dari data citra uji yang di-*testing* dengan menggunakan metode SVM diperoleh hasil seperti ditunjukkan pada Tabel 4.2

Tabel 4.2. Hasil Klasifikasi dengan Metode SVM dan Nilai Akurasi

Data uji ke-	Hasil Klasifikasi		Akurasi
1	S = 2, SB =2, SS = 0	G = 4, GB = 2, GS = 1	$\frac{4}{5} \times 100\% = 80\%$
	Tidak dihitung = 1		
2	S = 3, SB = 3, SS = 0	G = 5, GB = 1, GS = 1	$\frac{4}{5} \times 100\% = 80\%$
	Tidak dihitung = 3		
3	S = 4, SB = 4, SS = 0	G = 3, GB = 1, GS = 0	$\frac{5}{5} \times 100\% = 100\%$
	Tidak dihitung = 2		
4	S = 85 SB = 5, SS = 0	G = 1, GB = 0, GS = 0	$\frac{8}{9} \times 100\% = 88,89\%$
	Tidak dihitung = 1		
5	S = 5, SB = 4, SS = 0	G = 5, GB = 0, GS = 3	$\frac{4}{7} \times 100\% = 57,14\%$
	Tidak dihitung = 3		
6	S = 3, SB = 2, SS = 1	G = 5, GB = 0, GS = 3	$\frac{5}{7} \times 100\% = 71,43\%$
	Tidak dihitung = 1		
7	S = 2, SB = 1, SS = 1	G = 7, GB = 4, GS = 1	$\frac{5}{7} \times 100\% = 71,43\%$
	Tidak dihitung = 2		
8	S = 2, SB = 1, SS= 1	G = 7, GB = 4, GS = 1	$\frac{5}{7} \times 100\% = 71,43\%$
	Tidak dihitung = 2		
9	S = 3, SB = 1, SS = 1	G = 5, GB = 3, GS = 2	$\frac{4}{7} \times 100\% = 57,14\%$
	Tidak dihitung = 2		
10	S = 2, SB = 1, SS = 1	G = 3, GB = 0, GS = 0	$\frac{1}{2} \times 100\% = 50\%$
	Tidak dihitung = 2		
11	S = 4, SB = 1, SS = 1	G = 4, GB = 2, GS = 2	$\frac{3}{6} \times 100\% = 50\%$
	Tidak dihitung = 2		
12	S = 2, SB = 0, SS = 2	G = 5, GB = 1, GS = 0	$\frac{1}{3} \times 100\% = 33,33\%$
	Tidak dihitung = 3		
13	S = 4, SB = 1, SS = 1	G = 5, GB = 3, GS = 2	$\frac{4}{7} \times 100\% = 57,14\%$
	Tidak dihitung = 2		
14	S = 3, SB = 1, SS = 2	G = 6, GB = 4, GS = 2	$\frac{5}{7} \times 100\% = 71,43\%$
	Tidak dihitung = 2		

Data uji ke-	Hasil Klasifikasi		Akurasi
15	S = 2, SB = 1, SS = 0	G = 7, GB = 5, GS = 1	$\frac{6}{7} \times 100\% = 85,71\%$
	Tidak dihitung = 2		
16	S = , SB = , SS = 0	G = 7, GB = 6, GS = 0	$\frac{6}{6} \times 100\% = 100\%$
	Tidak dihitung = 1		
17	S = 1, SB = 1, SS = 0	G = 5, GB = 4, GS = 0	$\frac{5}{5} \times 100\% = 100\%$
	Tidak dihitung = 1		
18	S = 2, SB = 2, SS = 0	G = 5, GB = 5, GS = 0	$\frac{7}{7} \times 100\% = 100\%$
	Tidak dihitung = 0		
19	S = 1, SB = 0, SS = 0	G = 5, GB = 4, GS = 1	$\frac{4}{5} \times 100\% = 80\%$
	Tidak dihitung = 1		
Total			1416,18%
Rata-rata			74,54%

Keterangan :

S = daun semangka

SB = daun semangka yang terklasifikasikan secara tepat sesuai dengan kelasnya

SS = daun semangka yang salah diklasifikasikan

G = daun gulma

GB = daun gulma yang terklasifikasikan secara tepat sesuai dengan kelasnya

GS = daun gulma yang salah diklasifikasikan

Interpretasi dari Tabel 4.2 dapat dilihat pada data uji 1 yang terdiri dari 2 semangka 4 gulma (Lampiran 6 Gambar 1). Terdapat pengenalan objek yang seharusnya 2 dan 3 daun tapi dikenali sebagai 1 daun. Namun karena kedua daun tersebut adalah daun semangka, maka daun tersebut didefinisikan sebagai daun semangka. Begitu pula untuk daun gulma, jika 2 daun gulma dikenali dengan 1 daun maka daun tersebut didefinisikan sebagai daun gulma. Jika terdapat pengenalan objek daun semangka dan gulma yang dikenali sebagai 1 daun, maka objek tersebut tidak dihitung dalam hasil klasifikasi. Hasil klasifikasi data uji 2 terdiri dari 3 daun semangka dan 5 daun gulma (Lampiran 6 Gambar 2). Terdapat pengenalan objek yang seharusnya 2 dan 3 daun tapi dikenali sebagai 1 daun.

Namun karena kedua daun tersebut adalah daun semangka, maka daun tersebut didefinisikan sebagai daun semangka. Begitu pula untuk daun gulma, jika 2 daun gulma dikenali dengan 1 daun maka daun tersebut didefinisikan sebagai daun gulma. Jika terdapat pengenalan objek daun semangka dan gulma yang dikenal sebagai 1 daun, maka objek tersebut tidak dihitung dalam hasil klasifikasi. Untuk data uji selanjutnya diinterpretasikan seperti interpretasi sebelumnya.

Selanjutnya dilakukan perhitungan untuk mencari nilai akurasi setiap data uji dengan menggunakan rumus:

$$\text{Akurasi} = \frac{SB+GB}{SB+SS+GB+GS} \times 100\%$$

Hasil dari klasifikasi dengan metode SVM yang ditunjukkan pada Tabel 4.2 didapatkan bahwa metode SVM terbukti memiliki akurasi yang baik untuk mengklasifikasikan tanaman berdaun menjari (dalam penelitian ini mengambil sampel daun semangka) dan gulma berdasarkan fitur bentuk dan tekstur pada citra multi daun dengan menggunakan kernel *polinomial* derajat 2. Hal ini ditunjukkan oleh rata-rata akurasi yang dimiliki yaitu sebesar 74,54%.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil dari pembahasan pada Bab 4, dapat ditarik kesimpulan sebagai berikut:

- a. Metode SVM dapat mengklasifikasikan tanaman berdaun menjari dan gulma berdasarkan fitur bentuk dan tekstur pada citra multi daun dengan menggunakan kernel *polinomial* derajat 2. Data *training* untuk proses klasifikasi tanaman berdaun menjari dan gulma pada citra multi daun adalah menggunakan citra tunggal daun.
- b. Hasil dari klasifikasi dengan metode SVM yang ditunjukkan menyatakan bahwa metode SVM terbukti memiliki akurasi yang baik untuk mengklasifikasikan tanaman semangka dan gulma berdasarkan fitur bentuk tekstur pada citra multi daun. Rata-rata akurasi yang dimiliki yaitu sebesar 74,54%.

#### **5.2 Saran**

Dengan melihat hasil yang dicapai pada penelitian ini, terdapat beberapa hal yang penulis sarankan untuk pengembangan penelitian selanjutnya yaitu:

- a. Pengambilan data citra dengan teknik yang tepat akan menghasilkan informasi yang lebih akurat.
- b. Pada klasifikasi menggunakan metode SVM agar mendapatkan hasil yang lebih baik sebaiknya merubah atau menambahkan parameter-parameter yang digunakan.

## DAFTAR PUSTAKA

- Arribas, J. I., Sánchez-Ferrero, G. V., dan Ruiz-Ruiz, G., Gómez-Gil, J. (2011), “Leaf Classification in Sunflower Crops by Computer Vision and Neural Networks”, *Computers and Electronics in Agriculture*, Vo. 78, No. 1, hal. 9–18.
- Bai, X., Cao, Z., Wang, Y., Yu, Z., Hu, Z., Zhang, X., dan Li, C. (2014), “Vegetation Segmentation Robust To Illumination Variants Based On Clustering and Morphology Modelling”, *Biosystem Engineering*, Vol. 125, hal. 80 – 97.
- Chau, A. L., Li, X, and Yu, W. (2014), “Support Vector Machine Classification for Large Datasets using Decision Tree and Fisher Linier Discriminant”, *Future Generation Computer Systems*, Vol. 36, hal. 57–65.
- Devi, V dan P, Pallavi. (2014), “Leaf Recognition Based on Feature Extraction and Zernike Moments”, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2.
- Du, J.X., Wang, X.F. dan Zhang, G.J. (2007), “Leaf Shape Based Plant Species Recognition”, *Application Mathematics and Computation*, Vol. 185 hal. 883–893.
- Eldahshan, K.A., Youssef, M.I., Masameer, E.H., and Mustafa, M.A. (2014), “Segmentation Famework on Digital Microscope Image for Acute Lymphoblastic Leukimia Diagnosis based on HSV Color Space”, *International Journal of Computer Application* (0975 – 8887), Vol. 90, No. 7.
- Gonzalez, R. C., dan Woods, R. E. (2008), *Digital Image Processing*, 3<sup>rd</sup> edisi, Pearson Education, Inc., New Jersey.
- Gwo, C. Y., Wei, C. h., dan Li, Y. (2013), Rotary Matching of Edge Feature for Leaf Recognition, *Computer and Electronics in Agriculture*, Vol. 91, hal. 124-134.
- Hamel, L. (2009), *Knowledge Discovery with Spport Vector Machine*, JohnWiley & Sons, Inc., Hoboken, New Jersey.
- Han, J., Kamber, M., dan Pei, J. (2012), *Data Mining Concept and Techniques*, 3<sup>rd</sup> edisi, Elsevier, Inc., USA.

- <http://fireflyforest.net/firefly/2006/07/24/buffalobur-nightshade-a-very-evil-plant/>  
(1 april 2015).
- <http://plants.usda.gov/core/profile?symbol=soro> (1 april 2015).
- Kadir, A., Nugroha, L.E., Susanto, A., dan Santosa, P. I. (2011), “Neural Network Application on Foliage Plant Identification”, *International Journal of Computer Applications* ,Vol. 29, No.9, hal. 15 - 22.
- Kalyoncu, C. dan Toygar, Ö. (2014), “Geometric leaf classification”, *Computation Vision and Image Understanding*.  
<http://dx.doi.org/10.1016/j.cviu.2014.11.001>
- Kulkarni, T.A dan Bhosale, D.S. (2014), “A Robust Morphological Analysis of Normal and Abnormal Leukemic Cells Populations in Acute Lymphoblastic Leukemia”, *Proceedings of National Conf. on Recent Innovations in Science Engineering & Technology*, Pune, India, ISBN:978-93-84209-19-3.
- Narayan, V dan Subbarayan, G. (2014), “An Optimal Feature Subset Selection Using GA for Leaf Classification”, *The Internation Arab Journal of Information Technology*, Vol. 11, No. 5, Hal. 477 – 451.
- Qian, H., Mao, Y., Xiang,W dan Wang, Z. (2010), “Recognitionof Human Activities Using SVM Multi-Class Classifier”, *Pattern Recognition Letter*, Vol. 31, Hal. 100 – 111.
- Ratnasari, Juwita. (2008), *Galeri Tanaman Hias Daun*, Penerbit: Penebar Swadaya, Jakarta.
- Vijayakumar, S., Wu, S. (1999), *Sequential Support Vector Classifier and Regression*, Prociding International Conference on Soft Computing (SOCO’99), Genoa, Italy, Hal. 610-619.
- Wahyudi, T., dkk. (2008), *Panduan Lengkap Kakao Manajemen Agribisnis dari Hulu hingga Hilir*, Penerbit: Penebar Swadaya, Jakarta.
- Wang, X. F., Huang, D. S., Du, J. X., Xu, H., dan Heutte, L. (2008), “Classification of Plant Leaf Images with Complicated *Background*”, *Applied Mathematics and Computation*, Vol. 205, hal. 916-926.
- Wu, S.G., Bao, F.S., Xu, E.Y., Wang, Y.X., Chang, Y.F., dan Xiang, Q.L. (2007), “A Leaf Recognition Algorithm for Plant Calssification Using Probabilistic Neural Network”, *Signal Processing and Information Technology, IEEE International Symposium*, hal. 11-16.



## LAMPIRAN 1

### EKSTRAKSI FITUR BENTUK

#### LAMPIRAN 1.1 EKSTRAKSI FITUR BENTUK SEMANGKA

No.	AR	FF	R	FN	RPD	RPPLF
1	0,391315	16,24229	4,705518	0,077465	0,07351	0,077465
2	0,453732	12,54194	4,13531	0,065034	0,062225	0,065034
3	0,563477	9,878062	3,84268	0,079252	0,081555	0,079252
4	0,557441	13,55567	4,877137	0,084163	0,085021	0,084163
5	0,568106	11,80605	4,200397	0,085831	0,081161	0,085831
6	0,448454	13,83065	4,484757	0,062979	0,060563	0,062979
7	0,518509	15,45873	4,889356	0,029721	0,029469	0,029721
8	0,663546	10,75834	4,293599	0,033671	0,033414	0,033671
9	0,678273	10,41341	4,257813	0,041559	0,041613	0,041559
10	0,618586	11,14637	4,196843	0,116232	0,116164	0,116232
11	0,57401	18,21	6,424659	0,102817	0,106596	0,102817
12	0,674124	18,75576	6,944611	0,101877	0,101246	0,101877
13	0,627945	17,85957	6,624882	0,111441	0,110317	0,111441
14	0,457824	10,3894	3,393455	0,038229	0,037454	0,038229
15	0,47038	8,510235	2,965944	0,026104	0,024879	0,026104
16	0,249586	8,922054	2,049913	0,004568	0,004042	0,004568
17	0,729137	10,21788	4,213058	0,041455	0,040997	0,041455
18	0,599154	9,60555	3,599022	0,029653	0,028177	0,029653
19	0,561653	15,1381	5,673261	0,11374	0,115453	0,11374
20	0,553054	14,88198	5,477433	0,088844	0,092313	0,088844
21	0,375319	5,702744	1,531838	0,015717	0,01475	0,015717
22	0,630936	14,25938	5,104929	0,02602	0,025808	0,02602
23	0,453348	11,03544	3,756045	0,027558	0,026228	0,027558
24	0,416627	23,5887	6,14219	0,03068	0,030037	0,03068
25	0,311435	7,590258	2,048561	0,007709	0,008036	0,007709
26	0,22425	26,7221	4,93007	0,042246	0,043782	0,042246
27	0,52802	16,29253	5,541438	0,03358	0,033273	0,03358
28	0,422932	5,682303	1,895542	0,001976	0,002174	0,001976
29	0,681701	5,935657	2,517548	0,01312	0,013189	0,01312
30	0,384099	16,31527	4,097982	0,011504	0,011147	0,011504
31	0,365931	28,45656	7,654682	0,034257	0,031987	0,034257
32	0,655744	11,56546	4,15684	0,030137	0,030505	0,030137
33	0,616512	11,32224	4,427325	0,050544	0,049272	0,050544

No.	AR	FF	R	FN	RPD	RPPLF
34	0,643233	7,141138	3,052134	0,062259	0,06396	0,062259
35	0,439813	9,932001	3,439366	0,060469	0,059701	0,060469
36	0,389375	16,24689	5,012056	0,055237	0,055874	0,055237
37	0,400509	25,26785	6,882696	0,039215	0,039294	0,039215
38	0,201987	15,14255	2,62585	0,009539	0,008895	0,009539
39	0,205475	20,45885	3,699081	0,012556	0,012002	0,012556
40	0,600896	6,723597	2,838664	0,056102	0,057323	0,056102
41	0,606122	17,77489	6,755805	0,057226	0,056066	0,057226
42	0,465156	18,22328	5,755833	0,051752	0,050822	0,051752
43	0,289457	37,47963	8,20151	0,063192	0,061694	0,063192
44	0,577105	13,61406	4,356633	0,055013	0,055831	0,055013
45	0,544903	18,56133	6,564536	0,054618	0,055936	0,054618
46	0,516307	17,68453	6,160945	0,065183	0,065093	0,065183
47	0,416685	16,25557	4,916968	0,067983	0,067829	0,067983
48	0,441572	13,44646	4,736424	0,075293	0,072328	0,075293
49	0,707272	17,42895	6,554692	0,094218	0,097865	0,094218
50	0,320655	28,99571	6,97558	0,045894	0,044801	0,045894
51	0,404098	12,83247	3,700941	0,009471	0,009316	0,009471
52	0,372329	9,975708	2,969571	0,013623	0,013201	0,013623
53	0,450514	22,1358	5,865996	0,038252	0,037602	0,038252
54	0,631145	9,430042	3,720072	0,022616	0,021568	0,022616
55	0,329888	30,41012	6,669583	0,028616	0,027343	0,028616
56	0,377394	24,78619	6,290388	0,028997	0,029194	0,028997
57	0,491242	9,505707	3,428173	0,042782	0,043006	0,042782
58	0,476743	11,16698	3,802357	0,039471	0,043193	0,039471
59	0,704189	6,270434	2,871481	0,053343	0,052926	0,053343
60	0,323509	10,88409	2,868544	0,007986	0,008787	0,007986
61	0,408357	33,75151	9,306079	0,091675	0,087757	0,091675
62	0,40745	18,52573	5,760315	0,066041	0,064744	0,066041
63	0,421599	18,69844	5,185321	0,025914	0,025021	0,025914
64	0,534772	14,58409	5,140653	0,050664	0,048441	0,050664
65	0,319593	11,40212	2,759907	0,011683	0,011662	0,011683
66	0,734607	12,26279	5,196244	0,042877	0,043079	0,042877
67	0,67822	10,79399	4,59364	0,046469	0,048405	0,046469
68	0,396389	23,58137	6,379409	0,026111	0,024708	0,026111
69	0,341383	23,01102	5,611339	0,014834	0,013644	0,014834
70	0,486497	16,59635	5,357911	0,052034	0,053724	0,052034
71	0,619604	11,97858	4,599219	0,067858	0,070098	0,067858
72	0,216415	19,13935	3,40746	0,032617	0,031363	0,032617
73	0,50098	15,02827	4,589002	0,055946	0,056991	0,055946
74	0,581057	24,36145	7,542312	0,092592	0,092872	0,092592

No.	AR	FF	R	FN	RPD	RPPLF
75	0,661452	5,440216	2,449024	0,056997	0,056857	0,056997
76	0,676758	9,90872	4,184697	0,0422	0,043215	0,0422
77	0,569178	16,65675	5,85146	0,051001	0,052662	0,051001
78	0,549553	14,03776	5,003998	0,036212	0,037091	0,036212
79	0,575591	12,99794	4,672565	0,021454	0,020915	0,021454
80	0,679785	12,16235	4,731163	0,053295	0,053909	0,053295
81	0,327886	22,41739	5,901503	0,032452	0,0322	0,032452
82	0,751295	11,07231	4,751573	0,033449	0,034215	0,033449
83	0,555505	10,65795	4,164747	0,042823	0,041738	0,042823
84	0,539231	11,03996	4,242909	0,062856	0,063717	0,062856
85	0,444883	11,26082	3,629283	0,059825	0,061723	0,059825
86	0,431289	12,47678	3,830151	0,025241	0,024673	0,025241
87	0,739989	8,389275	3,48866	0,029803	0,030659	0,029803
88	0,709599	13,02463	4,847692	0,075346	0,080431	0,075346
89	0,388763	23,89533	6,118403	0,047142	0,052411	0,047142
90	0,629118	10,27284	3,966365	0,109891	0,111101	0,109891
91	0,557278	13,37924	4,149742	0,118284	0,118818	0,118284
92	0,572754	11,53072	4,004181	0,032955	0,031087	0,032955
93	0,432684	10,78152	3,448444	0,008461	0,007844	0,008461
94	0,737553	13,46216	5,124469	0,089691	0,091924	0,089691
95	0,493416	12,57024	4,208733	0,037705	0,036683	0,037705
96	0,521903	12,07521	3,743406	0,024796	0,025149	0,024796
97	0,529578	14,29999	5,184795	0,071626	0,071653	0,071626
98	0,504007	12,75697	4,327885	0,054097	0,053989	0,054097
99	0,431065	18,53268	5,623456	0,048484	0,049	0,048484
100	0,618744	10,9133	3,897856	0,029241	0,028853	0,029241
101	0,363788	15,26465	4,511081	0,076878	0,076216	0,076878
102	0,416011	15,60455	4,939792	0,051147	0,051065	0,051147
103	0,728793	8,036603	3,685689	0,065927	0,067835	0,065927
104	0,468567	25,77349	6,59065	0,028867	0,028668	0,028867
105	0,330637	17,88157	4,502905	0,020382	0,021138	0,020382
106	0,525093	18,91817	5,896531	0,028192	0,027435	0,028192
107	0,474794	10,24621	3,650265	0,028041	0,026792	0,028041
108	0,511518	18,82285	6,13734	0,044763	0,045261	0,044763
109	0,495376	14,09299	4,656994	0,060089	0,059864	0,060089
110	0,389119	28,255	7,938494	0,036271	0,03687	0,036271
111	0,671064	6,201788	2,710733	0,073236	0,072084	0,073236
112	0,468682	13,25726	4,480953	0,100303	0,10085	0,100303
113	0,645733	12,2175	4,393324	0,07572	0,075634	0,07572
114	0,525749	11,24216	3,935837	0,116803	0,119128	0,116803
115	0,454113	16,0824	5,036951	0,042142	0,040565	0,042142

No.	AR	FF	R	FN	RPD	RPPLF
116	0,565387	10,51526	4,042035	0,057163	0,056199	0,057163
117	0,467002	11,1822	3,650767	0,066199	0,065984	0,066199
118	0,494988	9,81545	3,329219	0,004384	0,004217	0,004384
119	0,62407	9,828573	3,817242	0,009176	0,009274	0,009176
120	0,43611	18,63043	5,158799	0,019274	0,018404	0,019274
121	0,775828	8,529815	3,838516	0,033921	0,035002	0,033921
122	0,302166	20,12354	4,095717	0,02154	0,021207	0,02154
123	0,665709	17,47056	6,485123	0,081988	0,080366	0,081988
124	0,385435	21,27946	6,098531	0,051243	0,049946	0,051243
125	0,487535	18,22082	5,571622	0,073496	0,073977	0,073496
126	0,636741	9,773712	3,804908	0,068369	0,067492	0,068369
127	0,673951	13,30727	5,221199	0,035045	0,036925	0,035045
128	0,63439	9,694343	4,028626	0,093776	0,095206	0,093776
129	0,581649	12,75572	4,848173	0,108102	0,107191	0,108102
130	0,425954	24,46773	6,958872	0,040673	0,041683	0,040673
131	0,391806	30,90454	8,204203	0,062365	0,062122	0,062365
132	0,647866	10,13281	3,470101	0,007373	0,007769	0,007373
133	0,573447	13,25799	4,492135	0,067911	0,067254	0,067911
134	0,484441	14,02996	4,610118	0,040315	0,038783	0,040315
135	0,523885	12,12401	4,284946	0,042353	0,042276	0,042353
136	0,408655	18,12414	5,231714	0,051724	0,051505	0,051724
137	0,515016	11,57233	4,203296	0,071524	0,070973	0,071524
138	0,714808	8,275958	3,517016	0,045497	0,044774	0,045497
139	0,474242	19,86208	6,610303	0,08629	0,087696	0,08629
140	0,638917	12,22632	4,846266	0,074661	0,074694	0,074661
141	0,479206	10,29548	3,508426	0,066165	0,064972	0,066165
142	0,474318	36,44373	10,07587	0,039079	0,040702	0,039079
143	0,183125	10,49914	1,796723	0,010051	0,010323	0,010051
144	0,214991	19,48839	3,376027	0,031151	0,029325	0,031151
145	0,708519	14,4906	5,197185	0,067223	0,065415	0,067223
146	0,699493	9,317604	3,919526	0,089978	0,088288	0,089978
147	0,379167	30,6687	8,066714	0,061145	0,05625	0,061145
148	0,523881	15,07907	4,722926	0,119293	0,125078	0,119293
149	0,561619	10,97073	3,931585	0,087379	0,093367	0,087379
150	0,663328	19,17528	7,16547	0,118593	0,114742	0,118593
151	0,575183	9,281203	3,851671	0,102371	0,100114	0,102371
152	0,37604	31,76168	8,27416	0,070251	0,069163	0,070251
153	0,293875	28,12858	6,518266	0,050077	0,050895	0,050077
154	0,526381	11,27828	4,035519	0,103294	0,104354	0,103294
155	0,297289	20,14823	4,791451	0,046915	0,046313	0,046915
156	0,403977	27,93398	8,038808	0,108404	0,105633	0,108404

# LAMPIRAN 1.2 EKSTRAKSI FITUR BENTU GULMA

No.	AR	FF	R	FN	RPD	RPPLF
1	0,74	9,819	3,85	0,076	0,075	0,076
2	0,667	10,82	3,983	0,06	0,057	0,06
3	0,664	9,546	3,647	0,063	0,061	0,063
4	0,712	11,09	4,154	0,067	0,064	0,067
5	0,716	11,02	4,136	0,062	0,061	0,062
6	0,739	9,502	3,758	0,043	0,042	0,043
7	0,734	10,02	3,853	0,056	0,056	0,056
8	0,724	10,53	3,952	0,04	0,039	0,04
9	0,755	9,384	3,711	0,068	0,064	0,068
10	0,749	9,384	3,625	0,063	0,063	0,063
11	0,702	10,09	3,893	0,048	0,049	0,048
12	0,685	8,355	3,146	0,044	0,044	0,044
13	0,631	8,853	3,281	0,049	0,047	0,049
14	0,687	9,403	3,503	0,044	0,044	0,044
15	0,737	6,509	2,638	0,026	0,027	0,026
16	0,674	10,9	3,887	0,045	0,045	0,045
17	0,68	11,99	4,438	0,04	0,039	0,04
18	0,649	10,71	3,94	0,09	0,093	0,09
19	0,689	10,73	3,957	0,06	0,062	0,06
20	0,687	8,694	3,278	0,055	0,053	0,055
21	0,778	8,289	3,577	0,057	0,053	0,057
22	0,705	7,062	2,876	0,064	0,063	0,064
23	0,707	8,64	3,433	0,064	0,063	0,064
24	0,702	9,939	3,89	0,062	0,062	0,062
25	0,739	8,897	3,354	0,061	0,065	0,061
26	0,732	8,578	3,314	0,039	0,04	0,039
27	0,691	10,65	4,052	0,043	0,043	0,043
28	0,671	8,874	3,424	0,054	0,05	0,054
29	0,753	8,718	3,471	0,06	0,062	0,06
30	0,667	10,26	3,68	0,032	0,032	0,032
31	0,699	9,383	3,416	0,038	0,038	0,038
32	0,709	9,463	3,76	0,045	0,044	0,045
33	0,679	8,48	3,381	0,048	0,049	0,048



## LAMPIRAN 2

### EKSTRAKSI FITUR TEKSTUR

#### LAMPIRAN 2.1 EKSTRAKSI FITUR TEKSTUR SEMANGKA

A. *Contrast* dan *Correlation* dengan sudut 0°, 45°, 90°, dan 135°.

No.	<i>Contrast</i>				<i>Correlation</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
1	0,077	0,074	0,077	0,074	0,126	0,17	0,126	0,17
2	0,065	0,062	0,065	0,062	0,131	0,167	0,131	0,167
3	0,079	0,082	0,079	0,082	0,164	0,139	0,164	0,139
4	0,084	0,085	0,084	0,085	0,153	0,144	0,153	0,144
5	0,086	0,081	0,086	0,081	0,122	0,169	0,122	0,169
6	0,063	0,061	0,063	0,061	0,136	0,169	0,136	0,169
7	0,03	0,029	0,03	0,029	0,19	0,196	0,19	0,196
8	0,034	0,033	0,034	0,033	0,179	0,184	0,179	0,184
9	0,042	0,042	0,042	0,042	0,18	0,178	0,18	0,178
10	0,116	0,116	0,116	0,116	0,126	0,126	0,126	0,126
11	0,103	0,107	0,103	0,107	0,157	0,125	0,157	0,125
12	0,102	0,101	0,102	0,101	0,138	0,143	0,138	0,143
13	0,111	0,11	0,111	0,11	0,128	0,136	0,128	0,136
14	0,038	0,037	0,038	0,037	0,162	0,178	0,162	0,178
15	0,026	0,025	0,026	0,025	0,158	0,197	0,158	0,197
16	0,005	0,004	0,005	0,004	0,147	0,244	0,147	0,244
17	0,041	0,041	0,041	0,041	0,168	0,177	0,168	0,177
18	0,03	0,028	0,03	0,028	0,148	0,19	0,148	0,19
19	0,114	0,115	0,114	0,115	0,14	0,127	0,14	0,127
20	0,089	0,092	0,089	0,092	0,164	0,131	0,164	0,131
21	0,016	0,015	0,016	0,015	0,164	0,216	0,164	0,216
22	0,026	0,026	0,026	0,026	0,179	0,184	0,179	0,184
23	0,028	0,026	0,028	0,026	0,186	0,225	0,186	0,225
24	0,031	0,03	0,031	0,03	0,183	0,2	0,183	0,2
25	0,008	0,008	0,008	0,008	0,239	0,207	0,239	0,207
26	0,042	0,044	0,042	0,044	0,191	0,161	0,191	0,161
27	0,034	0,033	0,034	0,033	0,164	0,172	0,164	0,172
28	0,002	0,002	0,002	0,002	0,268	0,194	0,268	0,194
29	0,013	0,013	0,013	0,013	0,184	0,179	0,184	0,179
30	0,012	0,011	0,012	0,011	0,188	0,212	0,188	0,212
31	0,034	0,032	0,034	0,032	0,162	0,217	0,162	0,217
32	0,03	0,031	0,03	0,031	0,184	0,173	0,184	0,173

No.	<i>Contrast</i>				<i>Correlation</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
33	0,051	0,049	0,051	0,049	0,16	0,181	0,16	0,181
34	0,062	0,064	0,062	0,064	0,17	0,147	0,17	0,147
35	0,06	0,06	0,06	0,06	0,159	0,168	0,159	0,168
36	0,055	0,056	0,055	0,056	0,177	0,166	0,177	0,166
37	0,039	0,039	0,039	0,039	0,191	0,188	0,191	0,188
38	0,01	0,009	0,01	0,009	0,189	0,243	0,189	0,243
39	0,013	0,012	0,013	0,012	0,203	0,238	0,203	0,238
40	0,056	0,057	0,056	0,057	0,175	0,156	0,175	0,156
41	0,057	0,056	0,057	0,056	0,164	0,18	0,164	0,18
42	0,052	0,051	0,052	0,051	0,165	0,179	0,165	0,179
43	0,063	0,062	0,063	0,062	0,164	0,183	0,164	0,183
44	0,055	0,056	0,055	0,056	0,185	0,172	0,185	0,172
45	0,055	0,056	0,055	0,056	0,189	0,168	0,189	0,168
46	0,065	0,065	0,065	0,065	0,161	0,161	0,161	0,161
47	0,068	0,068	0,068	0,068	0,158	0,159	0,158	0,159
48	0,075	0,072	0,075	0,072	0,139	0,172	0,139	0,172
49	0,094	0,098	0,094	0,098	0,157	0,123	0,157	0,123
50	0,046	0,045	0,046	0,045	0,185	0,204	0,185	0,204
51	0,009	0,009	0,009	0,009	0,192	0,205	0,192	0,205
52	0,014	0,013	0,014	0,013	0,187	0,212	0,187	0,212
53	0,038	0,038	0,038	0,038	0,181	0,195	0,181	0,195
54	0,023	0,022	0,023	0,022	0,168	0,207	0,168	0,207
55	0,029	0,027	0,029	0,027	0,192	0,228	0,192	0,228
56	0,029	0,029	0,029	0,029	0,204	0,198	0,204	0,198
57	0,043	0,043	0,043	0,043	0,184	0,179	0,184	0,179
58	0,039	0,043	0,039	0,043	0,219	0,144	0,219	0,144
59	0,053	0,053	0,053	0,053	0,168	0,173	0,168	0,173
60	0,008	0,009	0,008	0,009	0,254	0,179	0,254	0,179
61	0,092	0,088	0,092	0,088	0,137	0,173	0,137	0,173
62	0,066	0,065	0,066	0,065	0,155	0,171	0,155	0,171
63	0,026	0,025	0,026	0,025	0,202	0,228	0,202	0,228
64	0,051	0,048	0,051	0,048	0,154	0,19	0,154	0,19
65	0,012	0,012	0,012	0,012	0,222	0,223	0,222	0,223
66	0,043	0,043	0,043	0,043	0,171	0,166	0,171	0,166
67	0,046	0,048	0,046	0,048	0,193	0,159	0,193	0,159
68	0,026	0,025	0,026	0,025	0,179	0,222	0,179	0,222
69	0,015	0,014	0,015	0,014	0,177	0,242	0,177	0,242
70	0,052	0,054	0,052	0,054	0,179	0,151	0,179	0,151
71	0,068	0,07	0,068	0,07	0,177	0,149	0,177	0,149
72	0,033	0,031	0,033	0,031	0,169	0,2	0,169	0,2



No.	<i>Contrast</i>				<i>Correlation</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
73	0,056	0,057	0,056	0,057	0,183	0,167	0,183	0,167
74	0,093	0,093	0,093	0,093	0,157	0,154	0,157	0,154
75	0,057	0,057	0,057	0,057	0,169	0,17	0,169	0,17
76	0,042	0,043	0,042	0,043	0,19	0,169	0,19	0,169
77	0,051	0,053	0,051	0,053	0,188	0,16	0,188	0,16
78	0,036	0,037	0,036	0,037	0,193	0,173	0,193	0,173
79	0,021	0,021	0,021	0,021	0,177	0,196	0,177	0,196
80	0,053	0,054	0,053	0,054	0,171	0,161	0,171	0,161
81	0,032	0,032	0,032	0,032	0,193	0,198	0,193	0,198
82	0,033	0,034	0,033	0,034	0,194	0,175	0,194	0,175
83	0,043	0,042	0,043	0,042	0,166	0,186	0,166	0,186
84	0,063	0,064	0,063	0,064	0,168	0,155	0,168	0,155
85	0,06	0,062	0,06	0,062	0,183	0,157	0,183	0,157
86	0,025	0,025	0,025	0,025	0,191	0,208	0,191	0,208
87	0,03	0,031	0,03	0,031	0,181	0,157	0,181	0,157
88	0,075	0,08	0,075	0,08	0,178	0,122	0,178	0,122
89	0,047	0,052	0,047	0,052	0,219	0,132	0,219	0,132
90	0,11	0,111	0,11	0,111	0,135	0,125	0,135	0,125
91	0,118	0,119	0,118	0,119	0,132	0,128	0,132	0,128
92	0,033	0,031	0,033	0,031	0,164	0,211	0,164	0,211
93	0,008	0,008	0,008	0,008	0,175	0,234	0,175	0,234
94	0,09	0,092	0,09	0,092	0,154	0,132	0,154	0,132
95	0,038	0,037	0,038	0,037	0,166	0,187	0,166	0,187
96	0,025	0,025	0,025	0,025	0,193	0,18	0,193	0,18
97	0,072	0,072	0,072	0,072	0,152	0,15	0,152	0,15
98	0,054	0,054	0,054	0,054	0,167	0,168	0,167	0,168
99	0,048	0,049	0,048	0,049	0,178	0,168	0,178	0,168
100	0,029	0,029	0,029	0,029	0,166	0,176	0,166	0,176
101	0,077	0,076	0,077	0,076	0,147	0,153	0,147	0,153
102	0,051	0,051	0,051	0,051	0,165	0,166	0,165	0,166
103	0,066	0,068	0,066	0,068	0,177	0,152	0,177	0,152
104	0,029	0,029	0,029	0,029	0,206	0,21	0,206	0,21
105	0,02	0,021	0,02	0,021	0,232	0,204	0,232	0,204
106	0,028	0,027	0,028	0,027	0,202	0,223	0,202	0,223
107	0,028	0,027	0,028	0,027	0,193	0,229	0,193	0,229
108	0,045	0,045	0,045	0,045	0,194	0,184	0,194	0,184
109	0,06	0,06	0,06	0,06	0,166	0,168	0,166	0,168
110	0,036	0,037	0,036	0,037	0,201	0,187	0,201	0,187
111	0,073	0,072	0,073	0,072	0,148	0,162	0,148	0,162
112	0,1	0,101	0,1	0,101	0,142	0,137	0,142	0,137

No.	<i>Contrast</i>				<i>Correlation</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
113	0,076	0,076	0,076	0,076	0,152	0,152	0,152	0,152
114	0,117	0,119	0,117	0,119	0,151	0,133	0,151	0,133
115	0,042	0,041	0,042	0,041	0,166	0,196	0,166	0,196
116	0,057	0,056	0,057	0,056	0,164	0,178	0,164	0,178
117	0,066	0,066	0,066	0,066	0,157	0,16	0,157	0,16
118	0,004	0,004	0,004	0,004	0,208	0,238	0,208	0,238
119	0,009	0,009	0,009	0,009	0,21	0,2	0,21	0,2
120	0,019	0,018	0,019	0,018	0,175	0,211	0,175	0,211
121	0,034	0,035	0,034	0,035	0,202	0,176	0,202	0,176
122	0,022	0,021	0,022	0,021	0,196	0,208	0,196	0,208
123	0,082	0,08	0,082	0,08	0,141	0,158	0,141	0,158
124	0,051	0,05	0,051	0,05	0,173	0,194	0,173	0,194
125	0,073	0,074	0,073	0,074	0,166	0,16	0,166	0,16
126	0,068	0,067	0,068	0,067	0,157	0,167	0,157	0,167
127	0,035	0,037	0,035	0,037	0,205	0,163	0,205	0,163
128	0,094	0,095	0,094	0,095	0,156	0,143	0,156	0,143
129	0,108	0,107	0,108	0,107	0,14	0,146	0,14	0,146
130	0,041	0,042	0,041	0,042	0,207	0,187	0,207	0,187
131	0,062	0,062	0,062	0,062	0,162	0,165	0,162	0,165
132	0,007	0,008	0,007	0,008	0,248	0,207	0,248	0,207
133	0,068	0,067	0,068	0,067	0,155	0,163	0,155	0,163
134	0,04	0,039	0,04	0,039	0,173	0,204	0,173	0,204
135	0,042	0,042	0,042	0,042	0,179	0,179	0,179	0,179
136	0,052	0,052	0,052	0,052	0,165	0,167	0,165	0,167
137	0,072	0,071	0,072	0,071	0,155	0,161	0,155	0,161
138	0,045	0,045	0,045	0,045	0,178	0,19	0,178	0,19
139	0,086	0,088	0,086	0,088	0,163	0,149	0,163	0,149
140	0,075	0,075	0,075	0,075	0,163	0,163	0,163	0,163
141	0,066	0,065	0,066	0,065	0,153	0,169	0,153	0,169
142	0,039	0,041	0,039	0,041	0,186	0,15	0,186	0,15
143	0,01	0,01	0,01	0,01	0,222	0,2	0,222	0,2
144	0,031	0,029	0,031	0,029	0,154	0,203	0,154	0,203
145	0,067	0,065	0,067	0,065	0,143	0,166	0,143	0,166
146	0,09	0,088	0,09	0,088	0,143	0,159	0,143	0,159
147	0,061	0,056	0,061	0,056	0,154	0,221	0,154	0,221
148	0,119	0,125	0,119	0,125	0,154	0,115	0,154	0,115
149	0,087	0,093	0,087	0,093	0,168	0,111	0,168	0,111
150	0,119	0,115	0,119	0,115	0,119	0,147	0,119	0,147
151	0,102	0,1	0,102	0,1	0,139	0,157	0,139	0,157
152	0,07	0,069	0,07	0,069	0,168	0,181	0,168	0,181

No.	<i>Contrast</i>				<i>Correlation</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
153	0,05	0,051	0,05	0,051	0,192	0,179	0,192	0,179
154	0,103	0,104	0,103	0,104	0,14	0,131	0,14	0,131
155	0,047	0,046	0,047	0,046	0,176	0,185	0,176	0,185
156	0,108	0,106	0,108	0,106	0,137	0,159	0,137	0,159

B. *Energy* dan *Homogeneity* dengan sudut 0°, 45°, 90°, dan 135°.

No.	<i>Energy</i>				<i>Homogeneity</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
1	0,84	0,843	0,84	0,843	0,961	0,963	0,961	0,963
2	0,864	0,867	0,864	0,867	0,967	0,969	0,967	0,969
3	0,832	0,83	0,832	0,83	0,96	0,959	0,96	0,959
4	0,824	0,823	0,824	0,823	0,958	0,957	0,958	0,957
5	0,824	0,828	0,824	0,828	0,957	0,959	0,957	0,959
6	0,868	0,87	0,868	0,87	0,969	0,97	0,969	0,97
7	0,934	0,935	0,934	0,935	0,985	0,985	0,985	0,985
8	0,926	0,927	0,926	0,927	0,983	0,983	0,983	0,983
9	0,909	0,909	0,909	0,909	0,979	0,979	0,979	0,979
10	0,764	0,764	0,764	0,764	0,942	0,942	0,942	0,942
11	0,786	0,783	0,786	0,783	0,949	0,947	0,949	0,947
12	0,79	0,791	0,79	0,791	0,949	0,949	0,949	0,949
13	0,773	0,774	0,773	0,774	0,944	0,945	0,944	0,945
14	0,918	0,918	0,918	0,918	0,981	0,981	0,981	0,981
15	0,944	0,945	0,944	0,945	0,987	0,988	0,987	0,988
16	0,99	0,991	0,99	0,991	0,998	0,998	0,998	0,998
17	0,91	0,911	0,91	0,911	0,979	0,98	0,979	0,98
18	0,936	0,938	0,936	0,938	0,985	0,986	0,985	0,986
19	0,767	0,766	0,767	0,766	0,943	0,942	0,943	0,942
20	0,813	0,81	0,813	0,81	0,956	0,954	0,956	0,954
21	0,966	0,967	0,966	0,967	0,992	0,993	0,992	0,993
22	0,943	0,943	0,943	0,943	0,987	0,987	0,987	0,987
23	0,939	0,941	0,939	0,941	0,986	0,987	0,986	0,987
24	0,933	0,933	0,933	0,933	0,985	0,985	0,985	0,985
25	0,982	0,982	0,982	0,982	0,996	0,996	0,996	0,996
26	0,907	0,906	0,907	0,906	0,979	0,978	0,979	0,978
27	0,927	0,928	0,927	0,928	0,983	0,983	0,983	0,983
28	0,995	0,995	0,995	0,995	0,999	0,999	0,999	0,999
29	0,971	0,971	0,971	0,971	0,993	0,993	0,993	0,993
30	0,974	0,975	0,974	0,975	0,994	0,994	0,994	0,994
31	0,926	0,928	0,926	0,928	0,983	0,984	0,983	0,984
32	0,934	0,934	0,934	0,934	0,985	0,985	0,985	0,985
33	0,892	0,893	0,892	0,893	0,975	0,975	0,975	0,975
34	0,867	0,865	0,867	0,865	0,969	0,968	0,969	0,968
35	0,871	0,872	0,871	0,872	0,97	0,97	0,97	0,97
36	0,881	0,88	0,881	0,88	0,972	0,972	0,972	0,972
37	0,914	0,914	0,914	0,914	0,98	0,98	0,98	0,98
38	0,979	0,979	0,979	0,979	0,995	0,996	0,995	0,996

No.	<i>Energy</i>				<i>Homogeneity</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
39	0,972	0,972	0,972	0,972	0,994	0,994	0,994	0,994
40	0,879	0,878	0,879	0,878	0,972	0,971	0,972	0,971
41	0,878	0,879	0,878	0,879	0,971	0,972	0,971	0,972
42	0,889	0,89	0,889	0,89	0,974	0,975	0,974	0,975
43	0,865	0,867	0,865	0,867	0,968	0,969	0,968	0,969
44	0,881	0,88	0,881	0,88	0,972	0,972	0,972	0,972
45	0,881	0,88	0,881	0,88	0,973	0,972	0,973	0,972
46	0,861	0,862	0,861	0,862	0,967	0,967	0,967	0,967
47	0,856	0,856	0,856	0,856	0,966	0,966	0,966	0,966
48	0,843	0,846	0,843	0,846	0,962	0,964	0,962	0,964
49	0,803	0,8	0,803	0,8	0,953	0,951	0,953	0,951
50	0,9	0,901	0,9	0,901	0,977	0,978	0,977	0,978
51	0,979	0,979	0,979	0,979	0,995	0,995	0,995	0,995
52	0,97	0,97	0,97	0,97	0,993	0,993	0,993	0,993
53	0,916	0,917	0,916	0,917	0,981	0,981	0,981	0,981
54	0,951	0,952	0,951	0,952	0,989	0,989	0,989	0,989
55	0,937	0,938	0,937	0,938	0,986	0,986	0,986	0,986
56	0,935	0,935	0,935	0,935	0,986	0,985	0,986	0,985
57	0,907	0,906	0,907	0,906	0,979	0,978	0,979	0,978
58	0,912	0,908	0,912	0,908	0,98	0,978	0,98	0,978
59	0,885	0,886	0,885	0,886	0,973	0,974	0,973	0,974
60	0,981	0,981	0,981	0,981	0,996	0,996	0,996	0,996
61	0,811	0,814	0,811	0,814	0,954	0,956	0,954	0,956
62	0,86	0,861	0,86	0,861	0,967	0,968	0,967	0,968
63	0,942	0,943	0,942	0,943	0,987	0,987	0,987	0,987
64	0,892	0,894	0,892	0,894	0,975	0,976	0,975	0,976
65	0,973	0,973	0,973	0,973	0,994	0,994	0,994	0,994
66	0,907	0,907	0,907	0,907	0,979	0,978	0,979	0,978
67	0,898	0,896	0,898	0,896	0,977	0,976	0,977	0,976
68	0,943	0,944	0,943	0,944	0,987	0,988	0,987	0,988
69	0,967	0,969	0,967	0,969	0,993	0,993	0,993	0,993
70	0,887	0,886	0,887	0,886	0,974	0,973	0,974	0,973
71	0,854	0,852	0,854	0,852	0,966	0,965	0,966	0,965
72	0,929	0,93	0,929	0,93	0,984	0,984	0,984	0,984
73	0,879	0,878	0,879	0,878	0,972	0,972	0,972	0,972
74	0,806	0,806	0,806	0,806	0,954	0,954	0,954	0,954
75	0,878	0,878	0,878	0,878	0,972	0,972	0,972	0,972
76	0,908	0,907	0,908	0,907	0,979	0,978	0,979	0,978
77	0,889	0,887	0,889	0,887	0,974	0,974	0,974	0,974
78	0,92	0,919	0,92	0,919	0,982	0,981	0,982	0,981

No.	<i>Energy</i>				<i>Homogeneity</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
79	0,953	0,953	0,953	0,953	0,989	0,99	0,989	0,99
80	0,885	0,885	0,885	0,885	0,973	0,973	0,973	0,973
81	0,928	0,929	0,928	0,929	0,984	0,984	0,984	0,984
82	0,926	0,925	0,926	0,925	0,983	0,983	0,983	0,983
83	0,908	0,909	0,908	0,909	0,979	0,979	0,979	0,979
84	0,866	0,865	0,866	0,865	0,969	0,968	0,969	0,968
85	0,87	0,869	0,87	0,869	0,97	0,969	0,97	0,969
86	0,944	0,945	0,944	0,945	0,987	0,988	0,987	0,988
87	0,935	0,934	0,935	0,934	0,985	0,985	0,985	0,985
88	0,839	0,834	0,839	0,834	0,962	0,96	0,962	0,96
89	0,895	0,89	0,895	0,89	0,976	0,974	0,976	0,974
90	0,775	0,774	0,775	0,774	0,945	0,944	0,945	0,944
91	0,76	0,759	0,76	0,759	0,941	0,941	0,941	0,941
92	0,929	0,93	0,929	0,93	0,984	0,984	0,984	0,984
93	0,981	0,982	0,981	0,982	0,996	0,996	0,996	0,996
94	0,812	0,811	0,812	0,811	0,955	0,954	0,955	0,954
95	0,919	0,92	0,919	0,92	0,981	0,982	0,981	0,982
96	0,945	0,945	0,945	0,945	0,988	0,987	0,988	0,987
97	0,849	0,849	0,849	0,849	0,964	0,964	0,964	0,964
98	0,884	0,884	0,884	0,884	0,973	0,973	0,973	0,973
99	0,895	0,895	0,895	0,895	0,976	0,975	0,976	0,975
100	0,937	0,937	0,937	0,937	0,985	0,986	0,985	0,986
101	0,839	0,84	0,839	0,84	0,962	0,962	0,962	0,962
102	0,89	0,89	0,89	0,89	0,974	0,974	0,974	0,974
103	0,858	0,857	0,858	0,857	0,967	0,966	0,967	0,966
104	0,936	0,936	0,936	0,936	0,986	0,986	0,986	0,986
105	0,953	0,953	0,953	0,953	0,99	0,989	0,99	0,989
106	0,937	0,938	0,937	0,938	0,986	0,986	0,986	0,986
107	0,938	0,939	0,938	0,939	0,986	0,987	0,986	0,987
108	0,902	0,901	0,902	0,901	0,978	0,977	0,978	0,977
109	0,871	0,872	0,871	0,872	0,97	0,97	0,97	0,97
110	0,92	0,919	0,92	0,919	0,982	0,982	0,982	0,982
111	0,846	0,847	0,846	0,847	0,963	0,964	0,963	0,964
112	0,793	0,792	0,793	0,792	0,95	0,95	0,95	0,95
113	0,841	0,841	0,841	0,841	0,962	0,962	0,962	0,962
114	0,759	0,758	0,759	0,758	0,942	0,94	0,942	0,94
115	0,909	0,911	0,909	0,911	0,979	0,98	0,979	0,98
116	0,878	0,879	0,878	0,879	0,971	0,972	0,971	0,972
117	0,86	0,86	0,86	0,86	0,967	0,967	0,967	0,967
118	0,99	0,99	0,99	0,99	0,998	0,998	0,998	0,998

No.	<i>Energy</i>				<i>Homogeneity</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
119	0,979	0,979	0,979	0,979	0,995	0,995	0,995	0,995
120	0,958	0,959	0,958	0,959	0,99	0,991	0,99	0,991
121	0,925	0,924	0,925	0,924	0,983	0,982	0,983	0,982
122	0,952	0,952	0,952	0,952	0,989	0,989	0,989	0,989
123	0,829	0,831	0,829	0,831	0,959	0,96	0,959	0,96
124	0,889	0,891	0,889	0,891	0,974	0,975	0,974	0,975
125	0,844	0,843	0,844	0,843	0,963	0,963	0,963	0,963
126	0,855	0,856	0,855	0,856	0,966	0,966	0,966	0,966
127	0,922	0,92	0,922	0,92	0,982	0,982	0,982	0,982
128	0,804	0,803	0,804	0,803	0,953	0,952	0,953	0,952
129	0,778	0,779	0,778	0,779	0,946	0,946	0,946	0,946
130	0,91	0,909	0,91	0,909	0,98	0,979	0,98	0,979
131	0,867	0,867	0,867	0,867	0,969	0,969	0,969	0,969
132	0,983	0,982	0,983	0,982	0,996	0,996	0,996	0,996
133	0,856	0,857	0,856	0,857	0,966	0,966	0,966	0,966
134	0,913	0,914	0,913	0,914	0,98	0,981	0,98	0,981
135	0,908	0,908	0,908	0,908	0,979	0,979	0,979	0,979
136	0,889	0,889	0,889	0,889	0,974	0,974	0,974	0,974
137	0,849	0,849	0,849	0,849	0,964	0,965	0,964	0,965
138	0,901	0,902	0,901	0,902	0,977	0,978	0,977	0,978
139	0,818	0,817	0,818	0,817	0,957	0,956	0,957	0,956
140	0,842	0,842	0,842	0,842	0,963	0,963	0,963	0,963
141	0,86	0,861	0,86	0,861	0,967	0,968	0,967	0,968
142	0,914	0,913	0,914	0,913	0,98	0,98	0,98	0,98
143	0,977	0,977	0,977	0,977	0,995	0,995	0,995	0,995
144	0,933	0,935	0,933	0,935	0,984	0,985	0,984	0,985
145	0,859	0,86	0,859	0,86	0,966	0,967	0,966	0,967
146	0,813	0,815	0,813	0,815	0,955	0,956	0,955	0,956
147	0,87	0,875	0,87	0,875	0,969	0,972	0,969	0,972
148	0,754	0,749	0,754	0,749	0,94	0,937	0,94	0,937
149	0,815	0,81	0,815	0,81	0,956	0,953	0,956	0,953
150	0,761	0,764	0,761	0,764	0,941	0,943	0,941	0,943
151	0,789	0,791	0,789	0,791	0,949	0,95	0,949	0,95
152	0,85	0,851	0,85	0,851	0,965	0,965	0,965	0,965
153	0,89	0,89	0,89	0,89	0,975	0,975	0,975	0,975
154	0,787	0,787	0,787	0,787	0,948	0,948	0,948	0,948
155	0,898	0,899	0,898	0,899	0,977	0,977	0,977	0,977
156	0,778	0,78	0,778	0,78	0,946	0,947	0,946	0,947

## LAMPIRAN 2.2 EKSTRAKSI FITUR TEKSTUR GULMA

*Contrast* dan *Correlation* dengan sudut 0°, 45°, 90°, dan 135°.

No.	<i>Contrast</i>				<i>Correlation</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
1	0,076	0,075	0,076	0,075	0,157	0,166	0,157	0,166
2	0,06	0,057	0,06	0,057	0,145	0,198	0,145	0,198
3	0,063	0,061	0,063	0,061	0,157	0,187	0,157	0,187
4	0,067	0,064	0,067	0,064	0,153	0,195	0,153	0,195
5	0,062	0,061	0,062	0,061	0,159	0,169	0,159	0,169
6	0,043	0,042	0,043	0,042	0,172	0,193	0,172	0,193
7	0,056	0,056	0,056	0,056	0,166	0,168	0,166	0,168
8	0,04	0,039	0,04	0,039	0,17	0,175	0,17	0,175
9	0,068	0,064	0,068	0,064	0,144	0,192	0,144	0,192
10	0,063	0,063	0,063	0,063	0,162	0,158	0,162	0,158
11	0,048	0,049	0,048	0,049	0,189	0,176	0,189	0,176
12	0,044	0,044	0,044	0,044	0,179	0,185	0,179	0,185
13	0,049	0,047	0,049	0,047	0,154	0,193	0,154	0,193
14	0,044	0,044	0,044	0,044	0,155	0,165	0,155	0,165
15	0,026	0,027	0,026	0,027	0,196	0,162	0,196	0,162
16	0,045	0,045	0,045	0,045	0,17	0,162	0,17	0,162
17	0,04	0,039	0,04	0,039	0,159	0,168	0,159	0,168
18	0,09	0,093	0,09	0,093	0,172	0,148	0,172	0,148
19	0,06	0,062	0,06	0,062	0,186	0,162	0,186	0,162
20	0,055	0,053	0,055	0,053	0,15	0,181	0,15	0,181
21	0,057	0,053	0,057	0,053	0,143	0,205	0,143	0,205
22	0,064	0,063	0,064	0,063	0,162	0,176	0,162	0,176
23	0,064	0,063	0,064	0,063	0,157	0,157	0,157	0,157
24	0,062	0,062	0,062	0,062	0,161	0,158	0,161	0,158
25	0,061	0,065	0,061	0,065	0,188	0,139	0,188	0,139
26	0,039	0,04	0,039	0,04	0,187	0,173	0,187	0,173
27	0,043	0,043	0,043	0,043	0,183	0,181	0,183	0,181
28	0,054	0,05	0,054	0,05	0,133	0,196	0,133	0,196
29	0,06	0,062	0,06	0,062	0,182	0,149	0,182	0,149
30	0,032	0,032	0,032	0,032	0,175	0,183	0,175	0,183
31	0,038	0,038	0,038	0,038	0,166	0,173	0,166	0,173
32	0,045	0,044	0,045	0,044	0,17	0,186	0,17	0,186
33	0,048	0,049	0,048	0,049	0,173	0,167	0,173	0,167



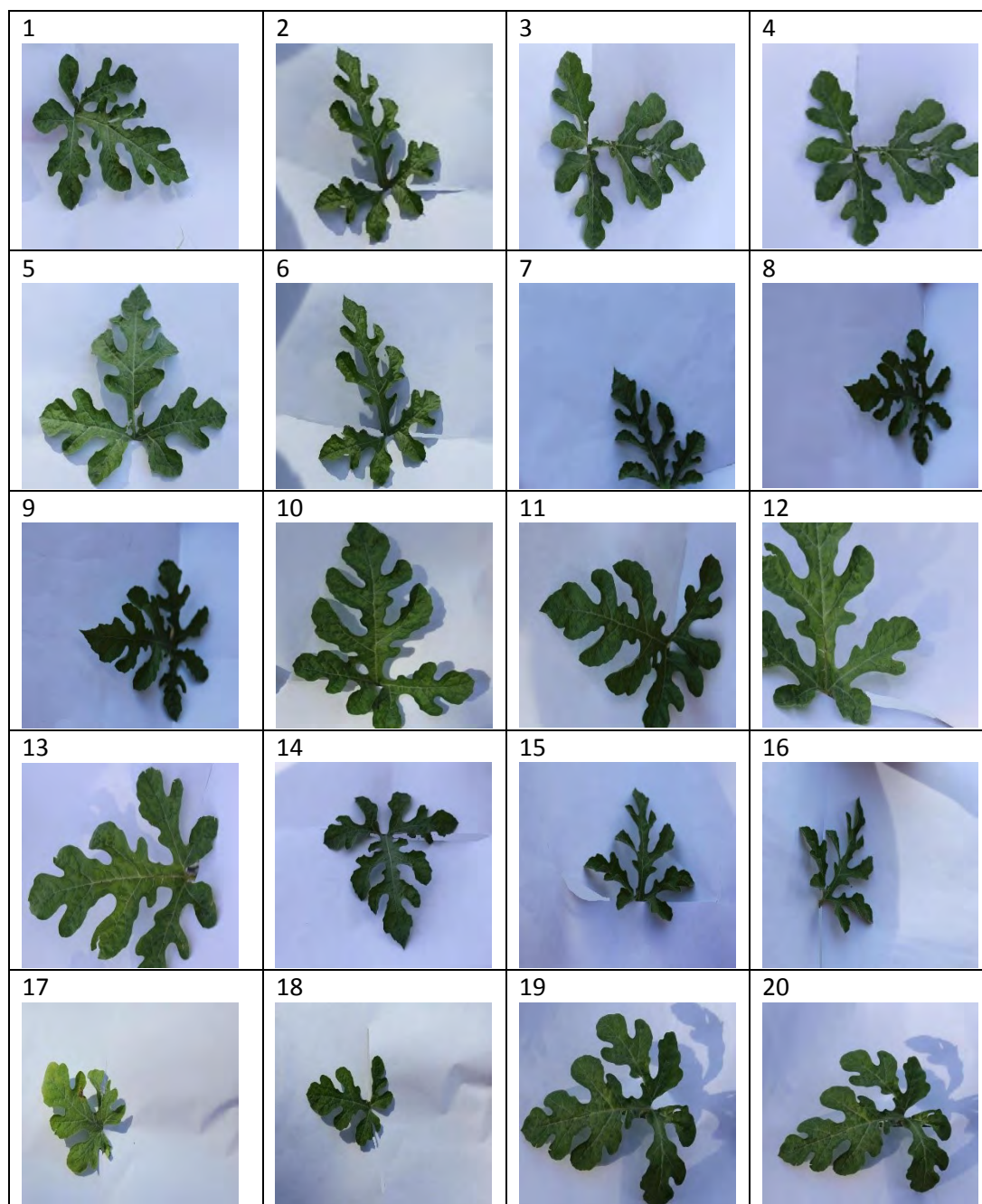
*Energy dan Homogeneity dengan sudut 0°, 45°, 90°, dan 135°.*

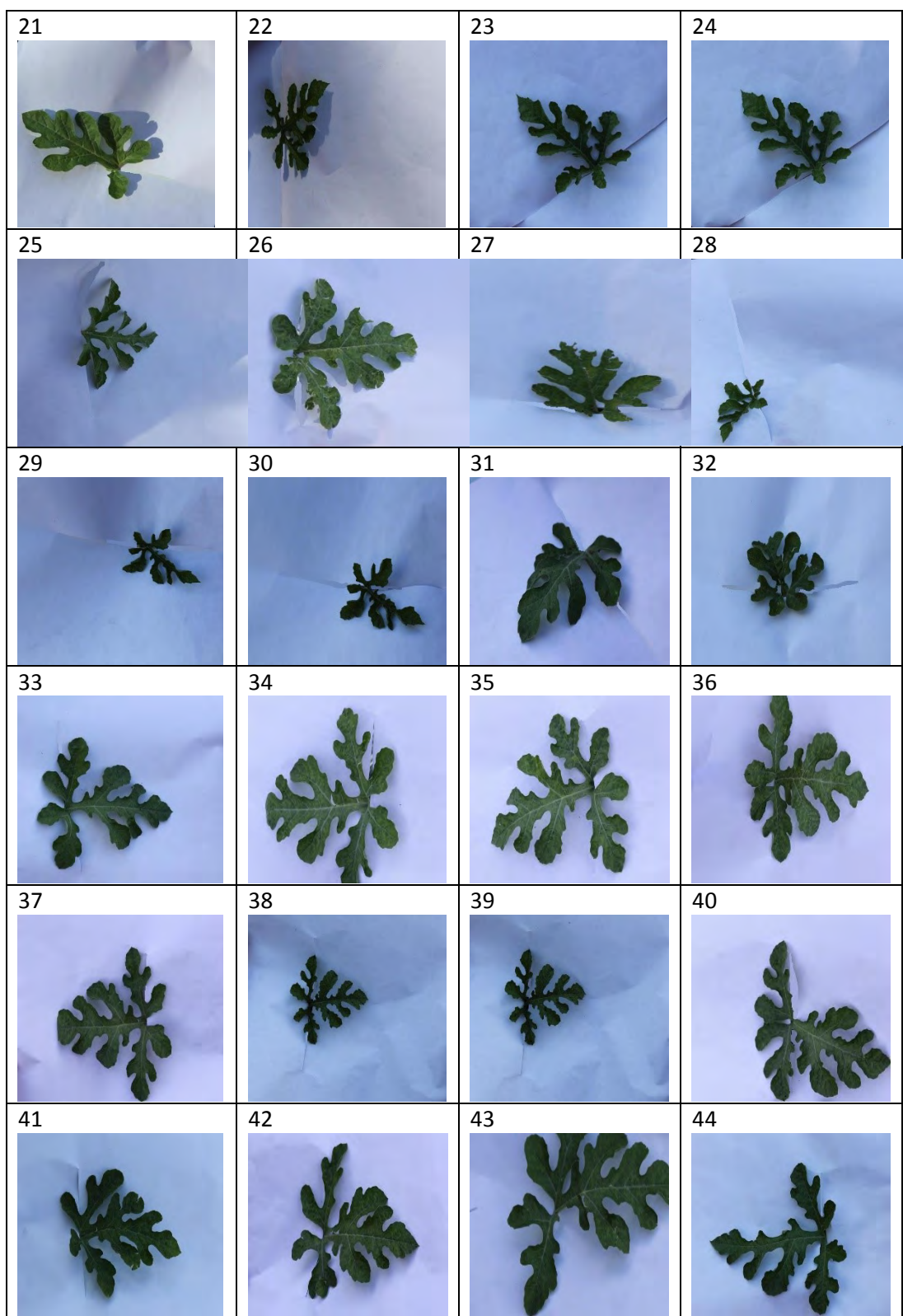
No.	<i>Energy</i>				<i>Homogeneity</i>			
	0°	45°	90°	135°	0°	45°	90°	135°
1	0,841	0,841	0,841	0,841	0,962	0,963	0,962	0,963
2	0,872	0,876	0,872	0,876	0,97	0,972	0,97	0,972
3	0,866	0,868	0,866	0,868	0,968	0,97	0,968	0,97
4	0,858	0,861	0,858	0,861	0,966	0,968	0,966	0,968
5	0,868	0,869	0,868	0,869	0,969	0,969	0,969	0,969
6	0,908	0,909	0,908	0,909	0,979	0,979	0,979	0,979
7	0,879	0,88	0,879	0,88	0,972	0,972	0,972	0,972
8	0,914	0,914	0,914	0,914	0,98	0,98	0,98	0,98
9	0,858	0,861	0,858	0,861	0,966	0,968	0,966	0,968
10	0,866	0,866	0,866	0,866	0,969	0,968	0,969	0,968
11	0,895	0,895	0,895	0,895	0,976	0,976	0,976	0,976
12	0,904	0,904	0,904	0,904	0,978	0,978	0,978	0,978
13	0,895	0,897	0,895	0,897	0,975	0,976	0,975	0,976
14	0,905	0,906	0,905	0,906	0,978	0,978	0,978	0,978
15	0,943	0,942	0,943	0,942	0,987	0,987	0,987	0,987
16	0,904	0,903	0,904	0,903	0,978	0,978	0,978	0,978
17	0,915	0,916	0,915	0,916	0,98	0,98	0,98	0,98
18	0,809	0,806	0,809	0,806	0,955	0,953	0,955	0,953
19	0,869	0,868	0,869	0,868	0,97	0,969	0,97	0,969
20	0,883	0,884	0,883	0,884	0,972	0,973	0,972	0,973
21	0,88	0,884	0,88	0,884	0,971	0,974	0,971	0,974
22	0,864	0,865	0,864	0,865	0,968	0,969	0,968	0,969
23	0,865	0,865	0,865	0,865	0,968	0,968	0,968	0,968
24	0,869	0,869	0,869	0,869	0,969	0,969	0,969	0,969
25	0,867	0,863	0,867	0,863	0,969	0,967	0,969	0,967
26	0,914	0,914	0,914	0,914	0,98	0,98	0,98	0,98
27	0,906	0,906	0,906	0,906	0,979	0,979	0,979	0,979
28	0,887	0,89	0,887	0,89	0,973	0,975	0,973	0,975
29	0,871	0,869	0,871	0,869	0,97	0,969	0,97	0,969
30	0,93	0,931	0,93	0,931	0,984	0,984	0,984	0,984
31	0,918	0,918	0,918	0,918	0,981	0,981	0,981	0,981
32	0,902	0,903	0,902	0,903	0,977	0,978	0,977	0,978
33	0,895	0,895	0,895	0,895	0,976	0,976	0,976	0,976

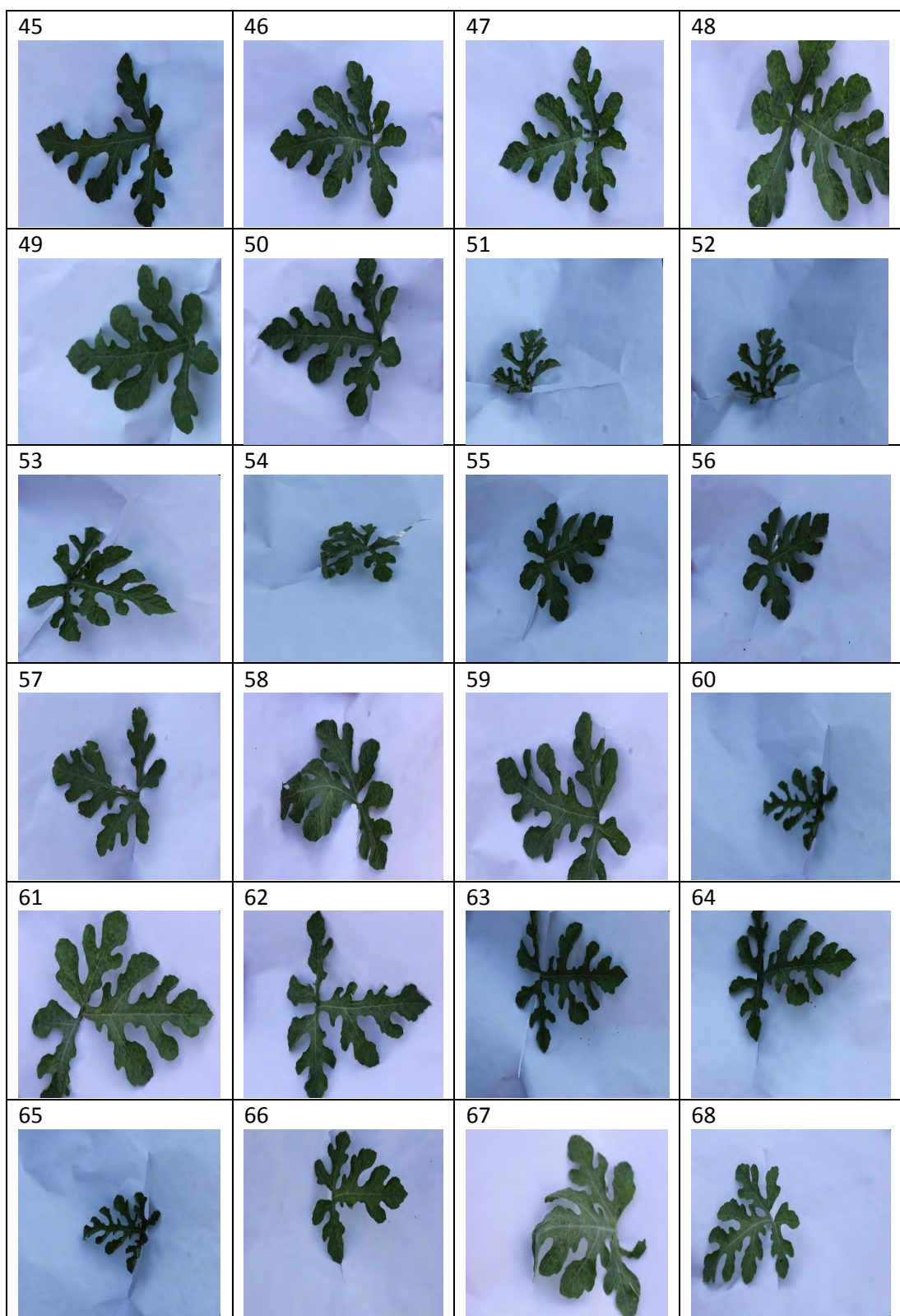


### LAMPIRAN 3

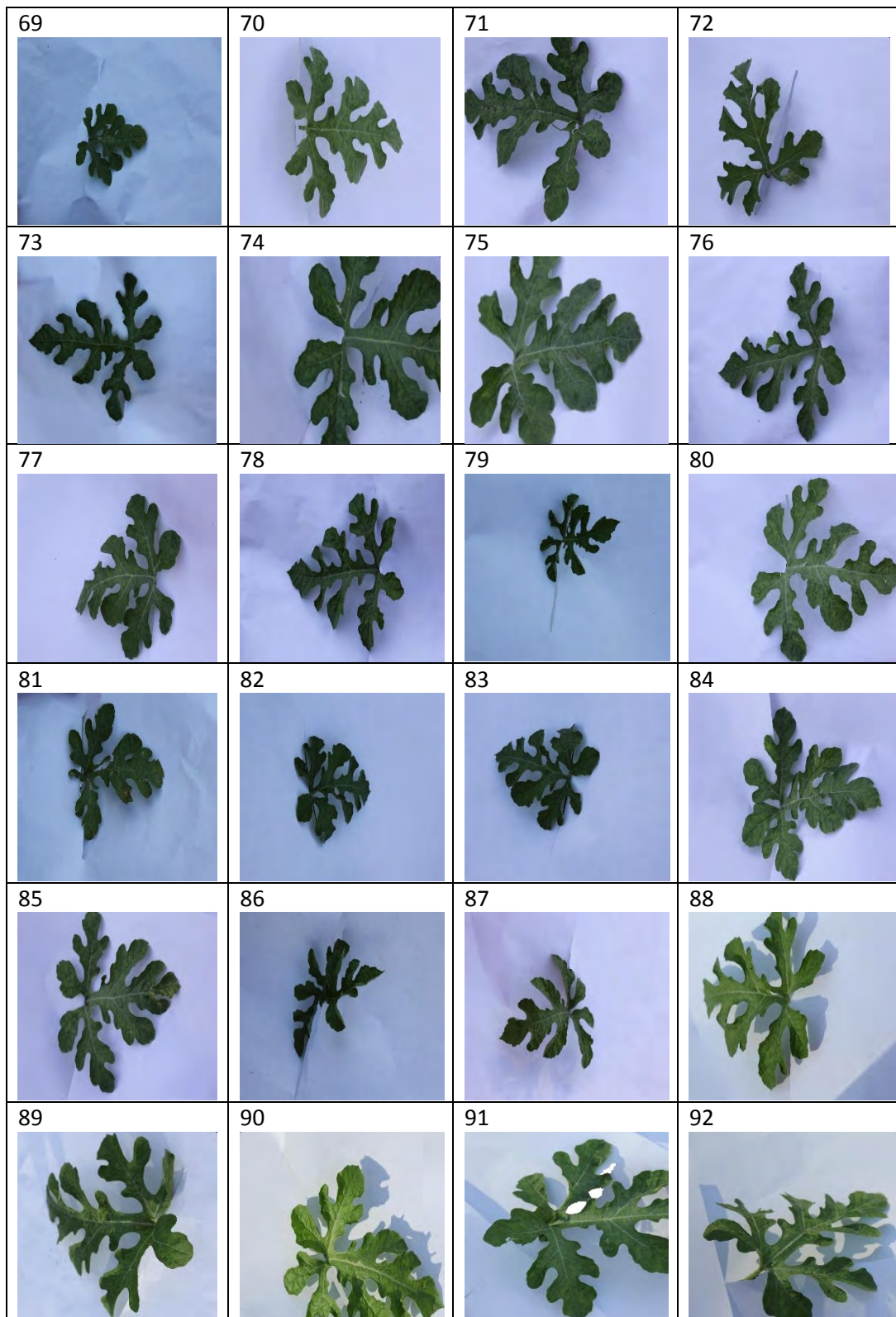
#### CITRA DAUN SEMANGKA

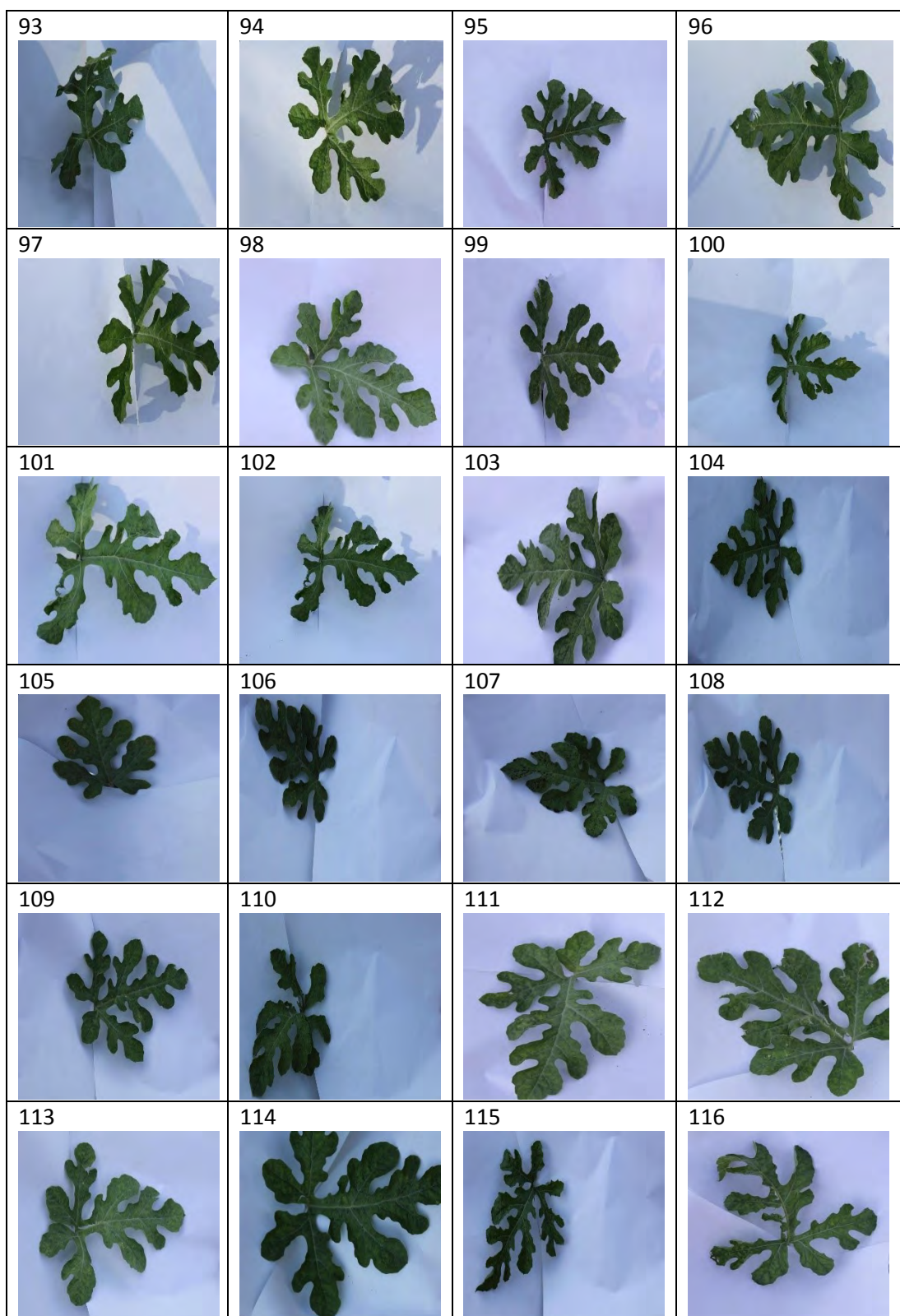




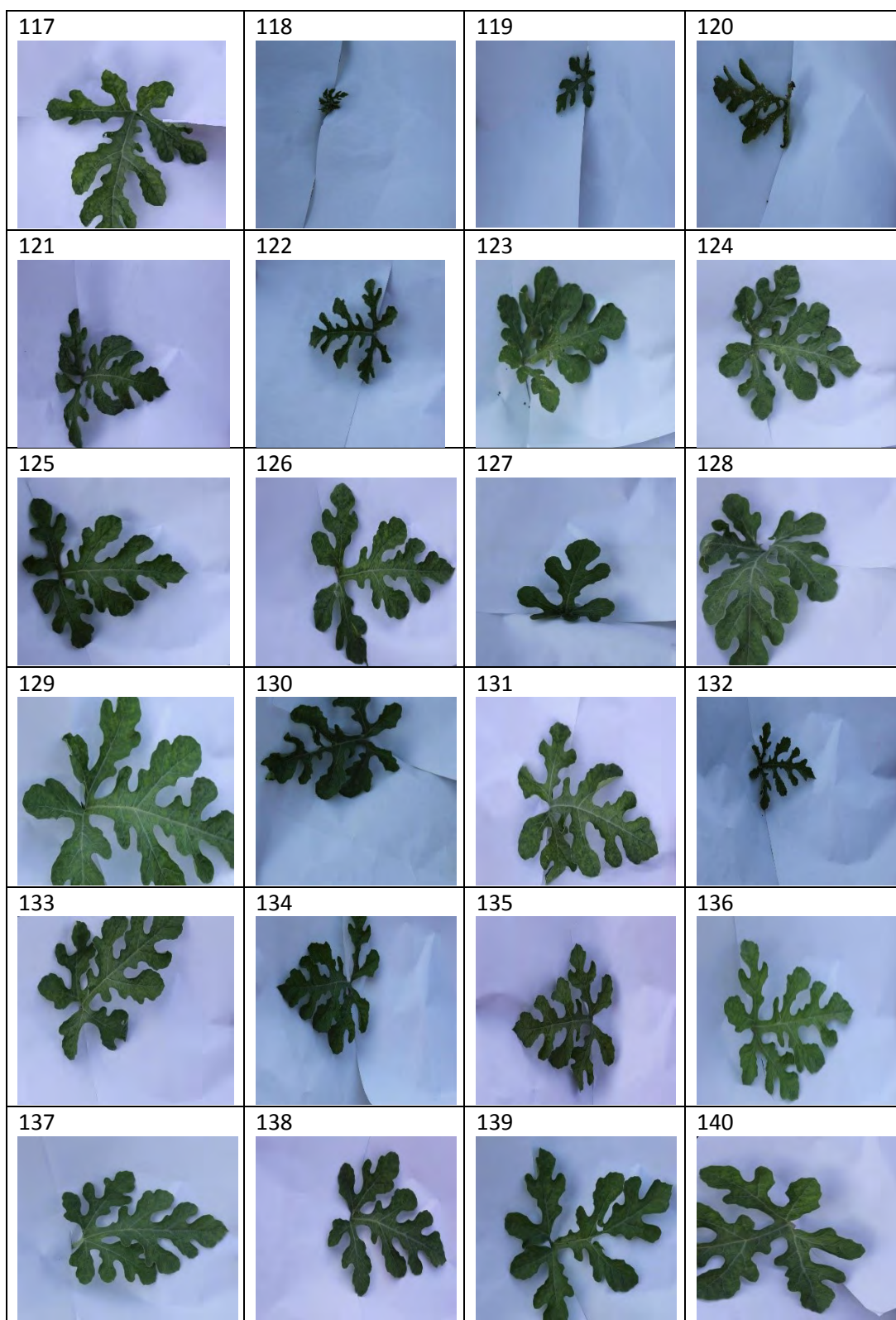




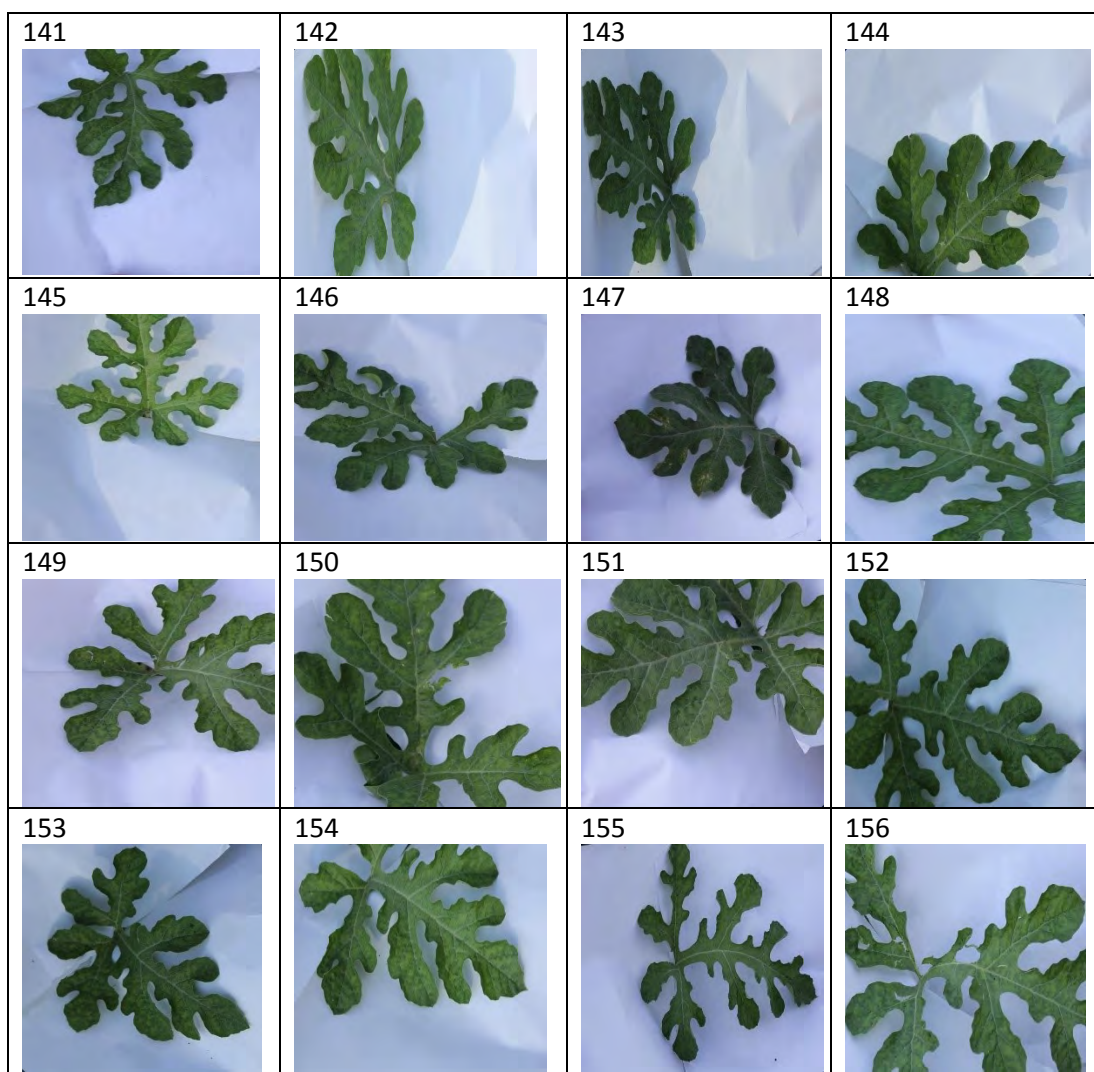






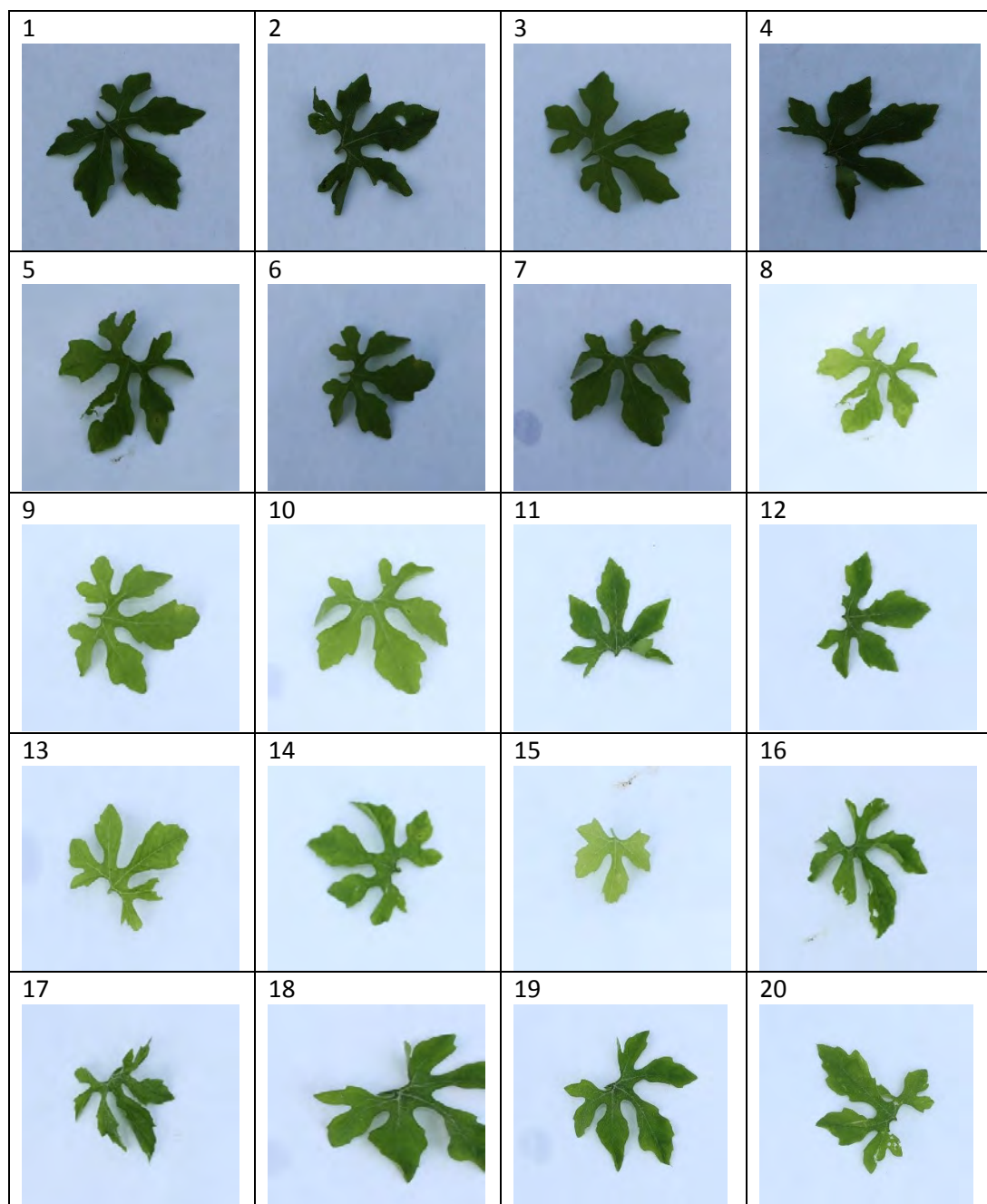


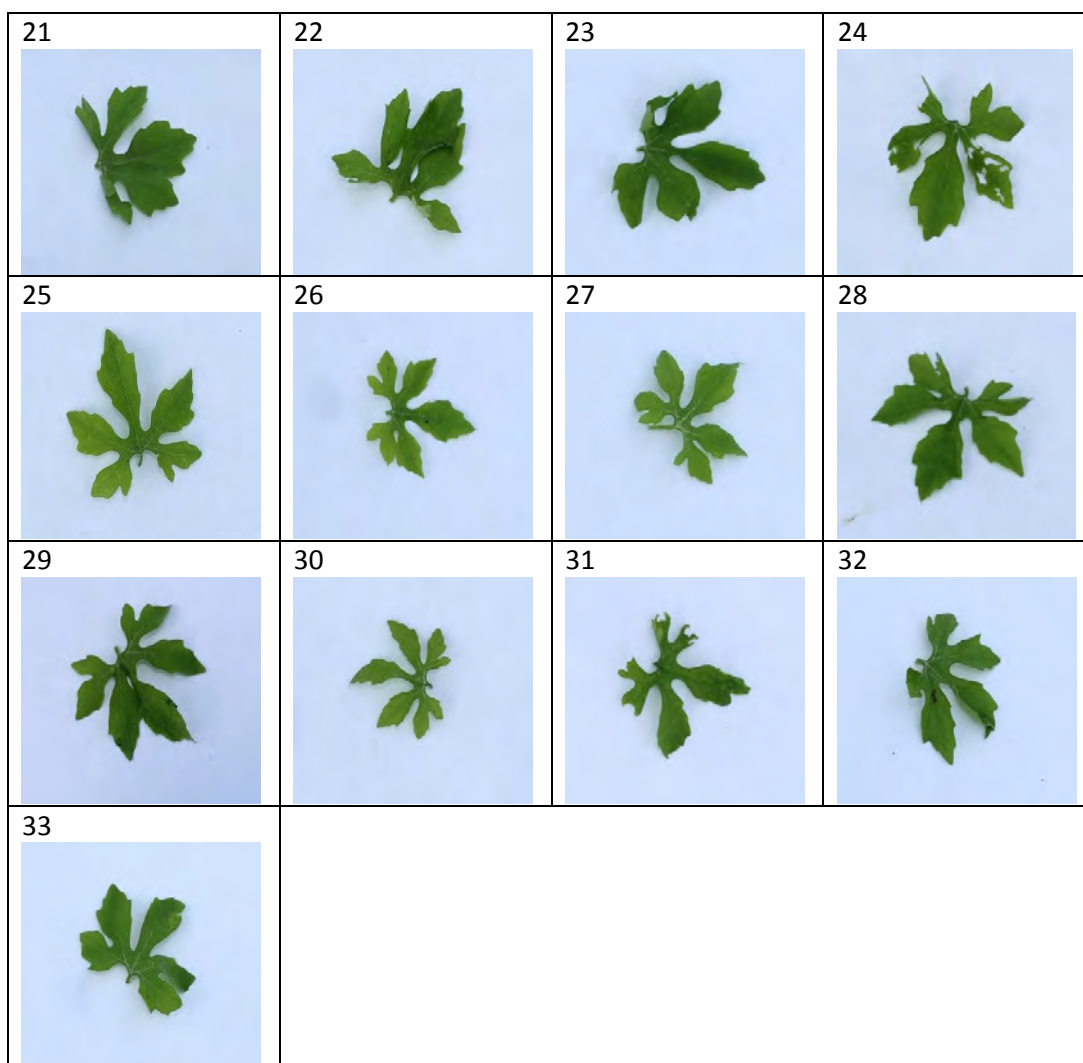




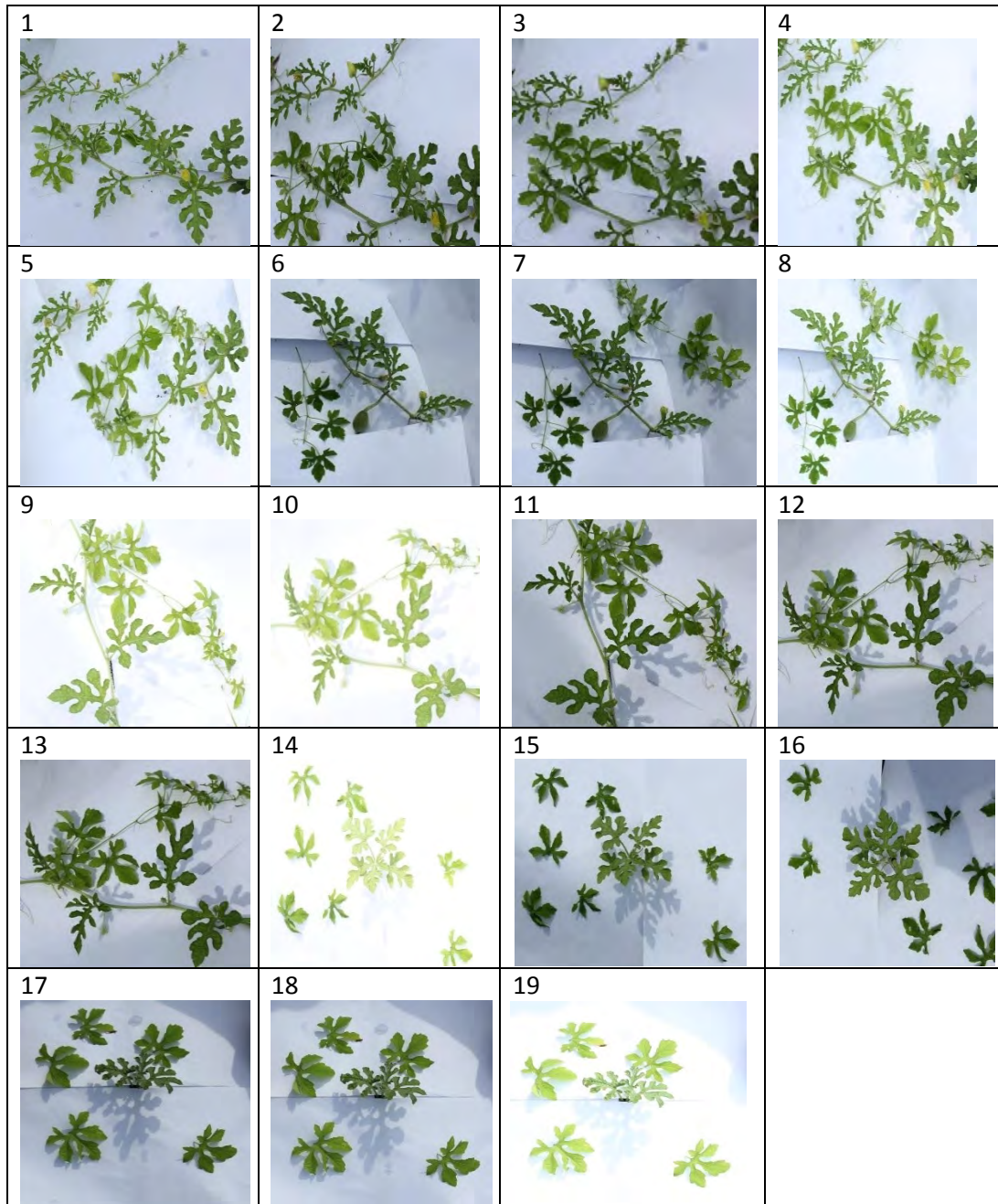


**LAMPIRAN 4**  
**CITRA DAUN GULMA**





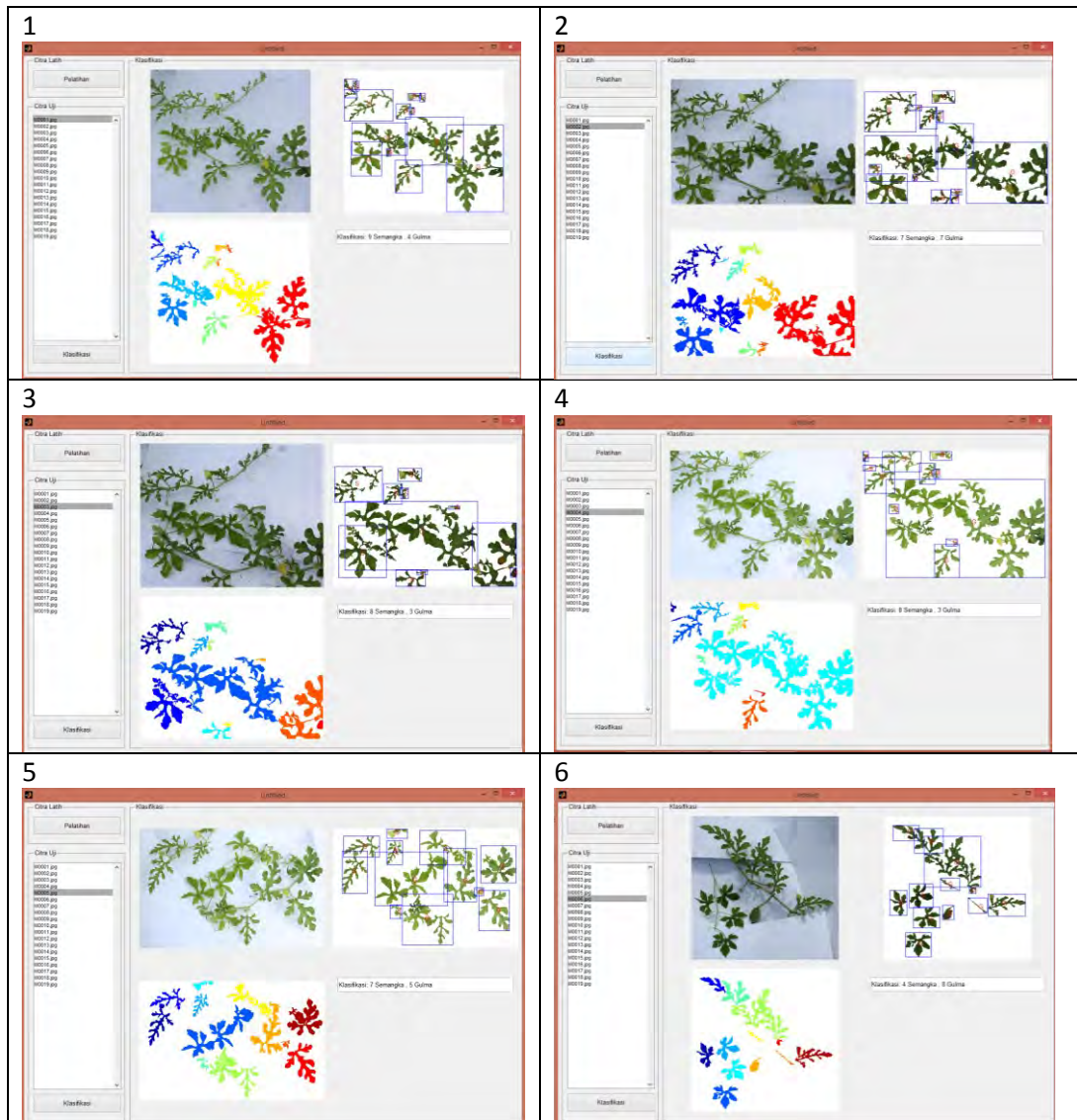
**LAMPIRAN 5**  
**CITRA MULTI DAUN**

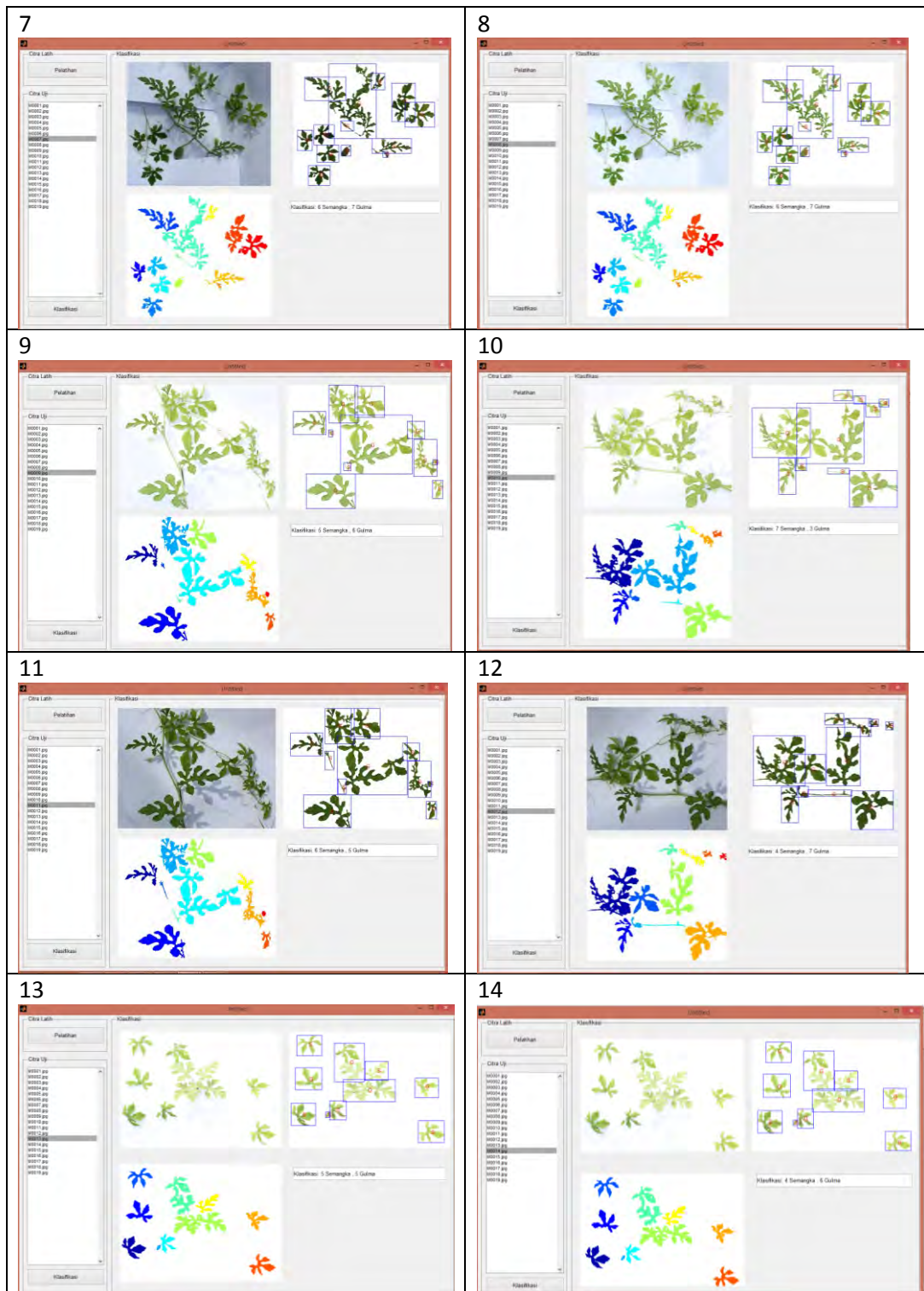






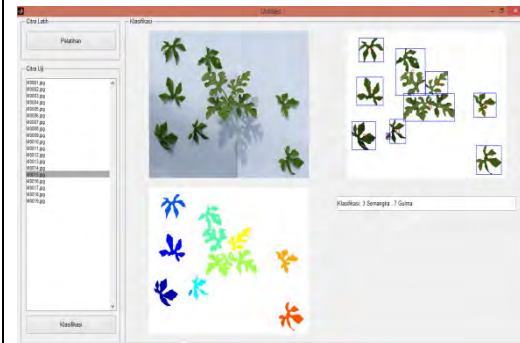
# **LAMPIRAN 6** **KLASIFIKASI MENGGUNAKAN** **METODE SUPPORT VECTOR MACHINE**



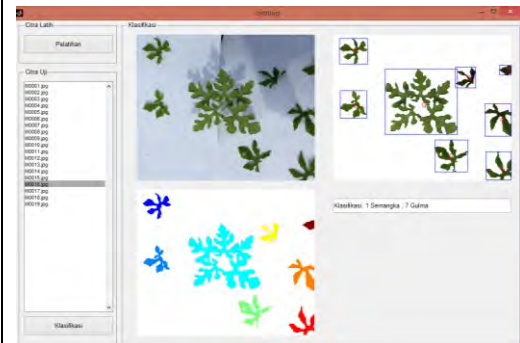




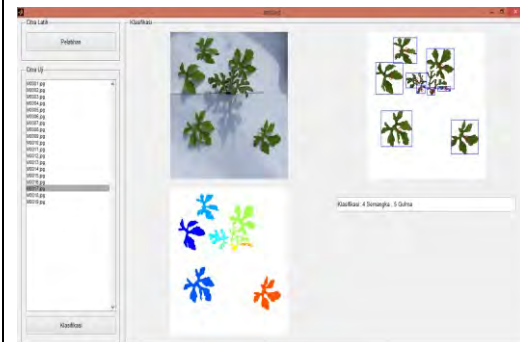
15



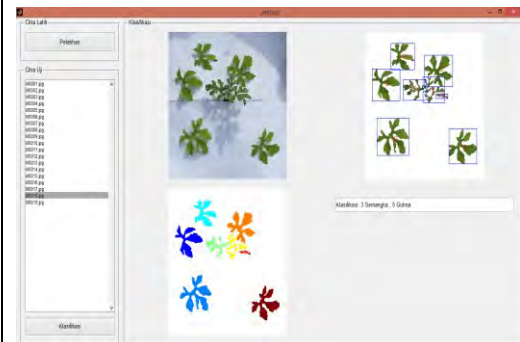
16



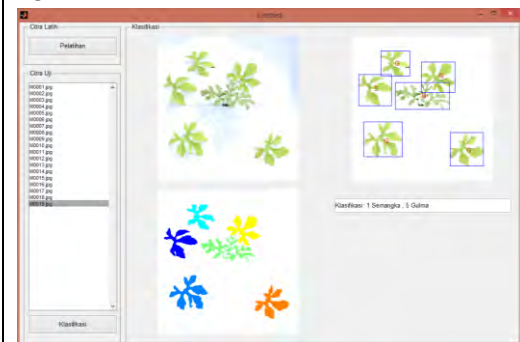
17



18



19





## BIODATA PENULIS



Penulis memiliki nama lengkap Etriana Meirista merupakan anak ketiga dari tiga bersaudara, lahir di Merauke pada tanggal 13 Mei 1988 dari pasangan Bapak H. Nasir dan Ibu (Almh) Hj. Setyaningsih. Penulis telah menempuh pendidikan formal mulai dari SD Negeri 2 Merauke, SLTP Negeri 1 Merauke, dan SMA Negeri 1 Merauke. Setelah lulus dari SMA, penulis melanjutkan studi S1 di Jurusan Pendidikan Matematika Fakultas Keguruan dan Ilmu Pendidikan Universitas Wisnuwardhana Malang dan diterima sebagai mahasiswa angkatan 2006. Penulis lulus sarjana dengan tujuh semester dan wisuda pada bulan Agustus 2010 dengan mendapat gelar Sarjana Pendidikan. Penulis menjadi salah satu dosen tidak tetap pada Universitas Musamus Merauke selama dua tahun sebelum akhirnya melanjutkan studi S2 di Jurusan Matematika Institut Teknologi Sepuluh Nopember (ITS) Surabaya melalui beasiswa Pra-S2 Matematika dengan kuliah matrikulasi selama 1 tahun pada tahun 2012, dilanjutkan dengan beasiswa Magister Matematika kuliah S2 pada tahun 2013 hingga tahun 2015 dengan NRP 1213201044. Selama kuliah S2 di jurusan matematika, penulis mengambil bidang komputasi. Kritik dan saran ataupun pertanyaan yang berhubungan dengan tesis ini dapat menghubungi penulis melalui email [etrianameirista47@gmail.com](mailto:etrianameirista47@gmail.com).